**实验二：FFT算法的MATLAB实现**

（一）实验目的：

理解离散傅立叶变换时信号分析与处理的一种重要变换，特别是FFT在数字信号处理中的高效率应用。

（二）实验原理：

1、有限长序列x(n)的DFT的概念和公式：





2、FFT算法

调用格式是

X= fft(x) 或 X=fft(x,N)

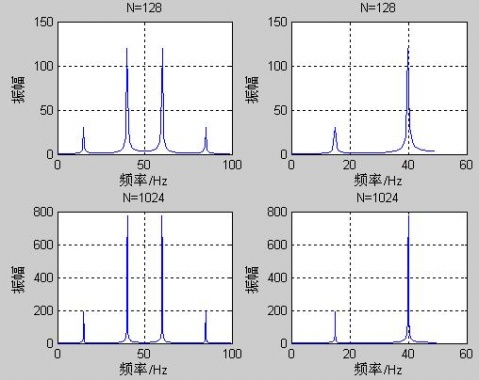
对前者，若x的长度是2的整数次幂，则按该长度实现x的快速变换，否则，实现的是慢速的非2的整数次幂的变换；对后者，N应为2的整数次幂，若x的长度小于N，则补零，若超过N，则舍弃N以后的数据。Ifft的调用格式与之相同。

**例1：**  
N=8;  
n=0:N-1;  
xn=[4 3 2 6 7 8 9 0];  
Xk=fft(xn)  
  
→  
Xk =

39.0000 -10.7782 + 6.2929i 0 - 5.0000i 4.7782 - 7.7071i 5.00004.7782 + 7.7071i0 + 5.0000i -10.7782 - 6.2929i

Xk与xn的维数相同，共有8个元素。

**例2：**x=0.5\*sin(2\*pi\*15\*t)+2\*sin(2\*pi\*40\*t)。采样频率fs=100Hz，分别绘制N=128、1024点幅频图。  
  
clf;  
fs=100;N=128;   %采样频率和数据点数  
n=0:N-1;t=n/fs;   %时间序列  
x=0.5\*sin(2\*pi\*15\*t)+2\*sin(2\*pi\*40\*t); %信号  
y=fft(x,N);    %对信号进行快速Fourier变换  
mag=abs(y);     %求得Fourier变换后的振幅  
f=n\*fs/N;    %频率序列  
subplot(2,2,1),plot(f,mag);   %绘出随频率变化的振幅  
xlabel('频率/Hz');  
ylabel('振幅');title('N=128');grid on;  
subplot(2,2,2),plot(f(1:N/2),mag(1:N/2)); %绘出Nyquist频率之前随频率变化的振幅  
xlabel('频率/Hz');  
ylabel('振幅');title('N=128');grid on;  
%对信号采样数据为1024点的处理  
fs=100;N=1024;n=0:N-1;t=n/fs;  
x=0.5\*sin(2\*pi\*15\*t)+2\*sin(2\*pi\*40\*t); %信号  
y=fft(x,N);   %对信号进行快速Fourier变换  
mag=abs(y);   %求取Fourier变换的振幅  
f=n\*fs/N;  
subplot(2,2,3),plot(f,mag); %绘出随频率变化的振幅  
xlabel('频率/Hz');  
ylabel('振幅');title('N=1024');grid on;  
subplot(2,2,4)  
plot(f(1:N/2),mag(1:N/2)); %绘出Nyquist频率之前随频率变化的振幅  
xlabel('频率/Hz');  
ylabel('振幅');title('N=1024');grid on;  
  
运行结果：



fs=100Hz，Nyquist频率为fs/2=50Hz。整个频谱图是以Nyquist频率为对称轴的。并且可以明显识别出信号中含有两种频率成分：15Hz和40Hz。由此可以知道FFT变换数据的对称性。因此用FFT对信号做谱分析，只需考察0~Nyquist频率范围内的福频特性。若没有给出采样频率和采样间隔，则分析通常对归一化频率0~1进行。另外，振幅的大小与所用采样点数有关，采用128点和1024点的相同频率的振幅是有不同的表现值，但在同一幅图中，40Hz与15Hz振动幅值之比均为4：1，与真实振幅0.5：2是一致的。为了与真实振幅对应，需要将变换后结果乘以2除以N。

（三）实验内容：

题一：若x(n)=cos(n\*pi/6)是一个N=12的有限序列，利用MATLAB计算它的DFT并画出图形。

题二：一被噪声污染的信号，很难看出它所包含的频率分量，如一个由50Hz和120Hz正弦信号构成的信号，受均值随机噪声的干扰，数据采样率为1000Hz，通过FFT来分析其信号频率成分，用MATLAB实现。

题三：调用原始语音信号mtlb，对其进行FFT变换后去掉幅值小于1的FFT变换值，最后重构语音信号。

（要求有四幅语音信号的频谱图在同一图形窗口以便比较：分别是1、原始语音信号；2、FFT变换；3去掉幅值小于1的FFT变换值；4、重构语音信号）

（四）实验报告：

题一：

clc;

N=12;

n=0:N-1;

xn=cos(n\*pi/6);

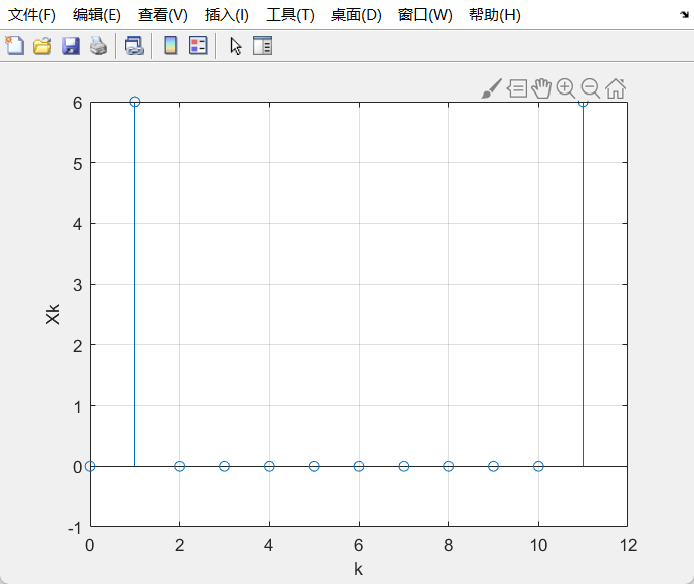
Xk=fft(xn,N);

Xk=abs(Xk);

stem(n,Xk);

xlabel('k');ylabel('Xk');

grid on;



题二：

clc;

fs=1000;

N=1024;

n=0:N-1;

t=n/fs;

f=n\*fs/N;

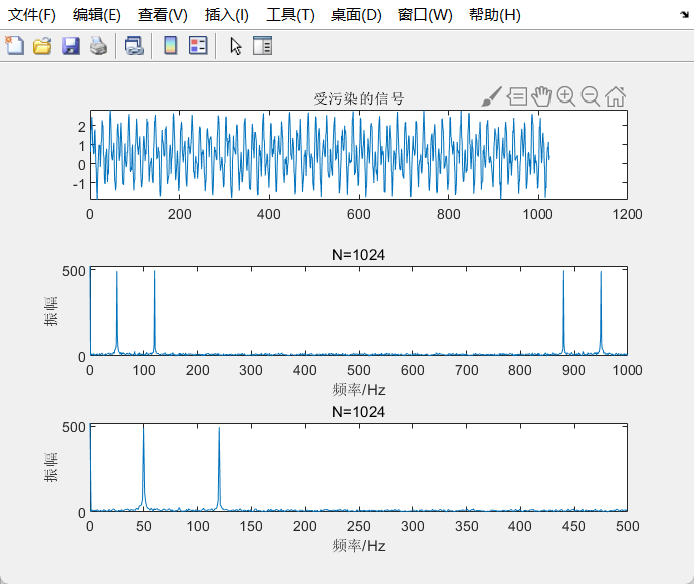
x=sin(2\*pi\*50\*t)+sin(2\*pi\*120\*t)+rand(1,N);

y=fft(x,N);

mag=abs(y);

subplot(1,2,1);plot(f,mag);xlabel('频率/Hz');ylabel('振幅');title('N=1024');

subplot(1,2,2);plot(f(1:N/2),mag(1:N/2));xlabel('频率/Hz');ylabel('振幅');title('N=1024');



题三：

clc;

load mtlb

N=4000;

subplot(2,3,1);plot([1:N],mtlb(1:N));title('原始语音信号');grid on;

y=fft(mtlb(1:N));

subplot(2,3,2);plot([1:N],abs(y));title('FFT变换');grid on;

y(abs(y)<1)=0;

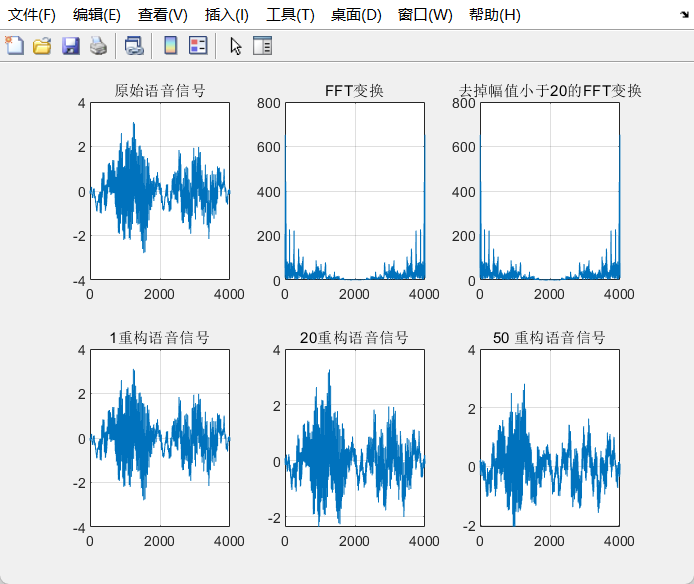
subplot(2,3,3);plot([1:N],abs(y));title('去掉幅值小于20的FFT变换');grid on;

subplot(2,3,4);plot([1:N],real(ifft(y)));title('1重构语音信号');grid on;

y(abs(y)<20)=0;

subplot(2,3,5);plot([1:N],real(ifft(y)));title('20重构语音信号');grid on;

y(abs(y)<50)=0;

subplot(2,3,6);plot([1:N],real(ifft(y)));title('50 重构语音信号');grid on;

**（五）心得体会**

通过此次实验，掌握了离散傅里叶变换的基本理论，特别是FFT的基本运算以及在数字信号处理中的广泛应用，并且知道了可以用FFT对信号进行分离和重构，同时还可以利用FFT来消除噪声对信号的干扰，对语音信号进行重构，从中可以看出傅里叶变换在DSP中的重要运用地位。