通信系统仿真
傅里叶变换函数

何晨光
哈尔滨工业大学
电子与信息工程学院

傅里叶变换函数

fft 函数

fftshift 函数

fftseg 函数

ifft 函数

FFT 函数

Y = fft(X)

用快速傅里叶变换 (FFT) 算法计算 X 的离散傅里叶变换 (DFT)。

Y = fft(X)

Y = fft(X,n)

信号X和采样点数N

FFT 函数

Y = fft(X)

如果 X 是向量，则 fft(X) 返回该向量的傅里叶变换。

如果 X 是矩阵，则 fft(X) 将 X 的各列视为向量，并返回每列的傅里叶变换。

如果 X 是一个多维数组，则 fft(X) 将沿大小不等于 1 的第一个数组维度的值视为向量，并返回每个向量的傅里叶变换。

FFT 函数

Y = fft(X,n)

信号X和采样点数，返回 n 点 DFT

如果未指定任何值，则 Y 的大小与 X 相同。

如果 X 是向量且 X 的长度小于 n，则为 X 补上尾零以达到长度 n。

如果 X 是向量且 X 的长度大于 n，则对 X 进行截断以达到长度 n。

FFT 函数

N值

N决定了频率采样的点数。

注意此采样点数不是时域的，因为时域的采样频率决定了频域的最大值（fmax=Fs），所以在已知道信号的时域采样率后我们就可是确认频率域的分辨率df，即df=Fs/N.

所以N可以根据分辨率的需要来自行设定，在没有输入N的情况下系统默认为信号时域的采样点数。但是注意设定的N值要产生的最小分辨率要小于被分析信号频率间的最小差值，要不然会失真。

FFT 函数

采样频率Fs

采样频率Fs，表示在Fs中间被N-1个点平均分成N等份，每个点的频率依次增加。

某点n所表示的频率为：Fn=(n-1)*Fs/N。

由上面的公式可以看出，Fn所能分辨到频率为df=Fs/N，如果采样频率Fs为1024Hz，采样点数为1024点，则可以分辨到1Hz。

FFT 函数

采样频率Fs

1024Hz的采样率：采样1024点，刚好是1秒，也就是说，采样1秒时间的信号并做FFT，则结果可以分析到1Hz，如果采样2秒时间的信号并做FFT，则结果可以分析到0.5Hz。如果要提高频率分辨力df=Fs/N，则必须增加采样点数，也即采样时间。

频率分辨率和采样时间是倒数关系。

FFT 函数

频率轴的处理

$$X(k)=\sum_{n=1}^{N}x(n)\exp(-j2\pi\,(k-1)(n-1)/N),\qquad 1\le k\le N$$

可知，采样得到的频率被分成了N份，范围是[0,Fs]，也就是[0,N]。所以fft得到的频域不是真实的频域，而是基于采样点数得到的，那么采样点的不同取值除了会使频率轴的取值有变化外，傅里叶变换的形状完全没有变化，当然是在N的取值不会使得频谱失真的情况下的。所以在满足不失真的情况下，不管N为多少，我们都可以得到准确的傅里叶变换图形，只是分辨率不同。但是要获得真正的频率轴，需要对频率轴进行归一化处理。

FFT 函数

频率轴的处理

所以在满足不失真的情况下，不管N为多少，我们都可以得到准确的傅里叶变换图形，只是分辨率不同。但是要获得真正的频率轴，需要对频率轴进行归一化处理。

在满足采样定律的情况下傅里叶变换得到的谱的最大频率为采样频率的一半Fs/2（Nyquist频率）,所以函数fft返回值是以Nyqusit频率为轴对称的，Y的前一半与后一半是复数共轭关系，只需要对频率轴做一个简单的变化即可，即频率都除以N值，然后截取前一半。

FFT 函数

频率轴的处理

X轴频率点的设置：

采样频率为Fs

频谱图显示的最高频率为Fs/2（采样定理）

X轴频率点：　（0:1：N/2）*Fs/N

FFT 函数

副值处理

MATLAB中FFT的频谱，应该看幅值。

假设采样频率为Fs，信号频率F，采样点数为N。那么FFT之后结果就是一个为N点的复数。每一个点就对应着一个频率点。这个点的模值，就是该频率值下的幅度特性。

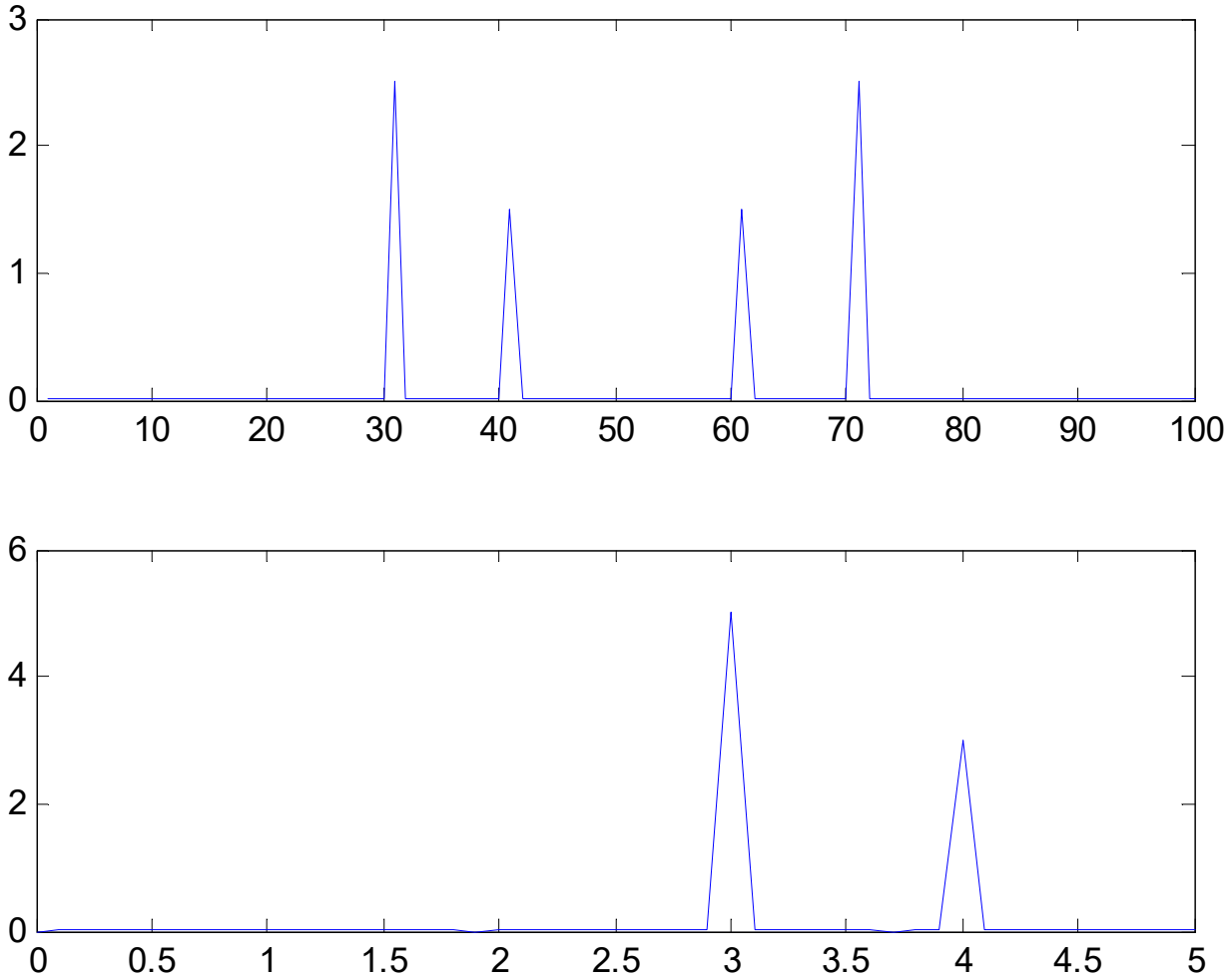
具体跟原始信号幅度的关系：假设原始信号的峰值为A，那么FFT的结果的每个点（除了第一个点直流分量之外）的模值就是A的N/2倍。而第一个点就是直流分量，它的模值就是直流分量的N倍。而每个点的相位呢，就是在该频率下的信号的相位。。

FFT 函数 例c0501.m

```
Fs = 10;                % 采样频率
T = 1/Fs;               % 采样间隔
L = 100;                % 信号长度
t = (0:L-1)*T;          % 时间序列

xt=5*cos(6*pi*t)+3*sin(8*pi*t);

Y1=fft(xt);

Y=abs(Y1/L);
P1 = Y(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);

f=Fs*(0:(L/2))/L;
subplot(211);plot(abs(Y1)/L);
subplot(212);plot(f,P1)
```

FFT 函数 例c0501.m

普通的坐标轴没有乘以Fs/N



FFT 函数 例c0502.m

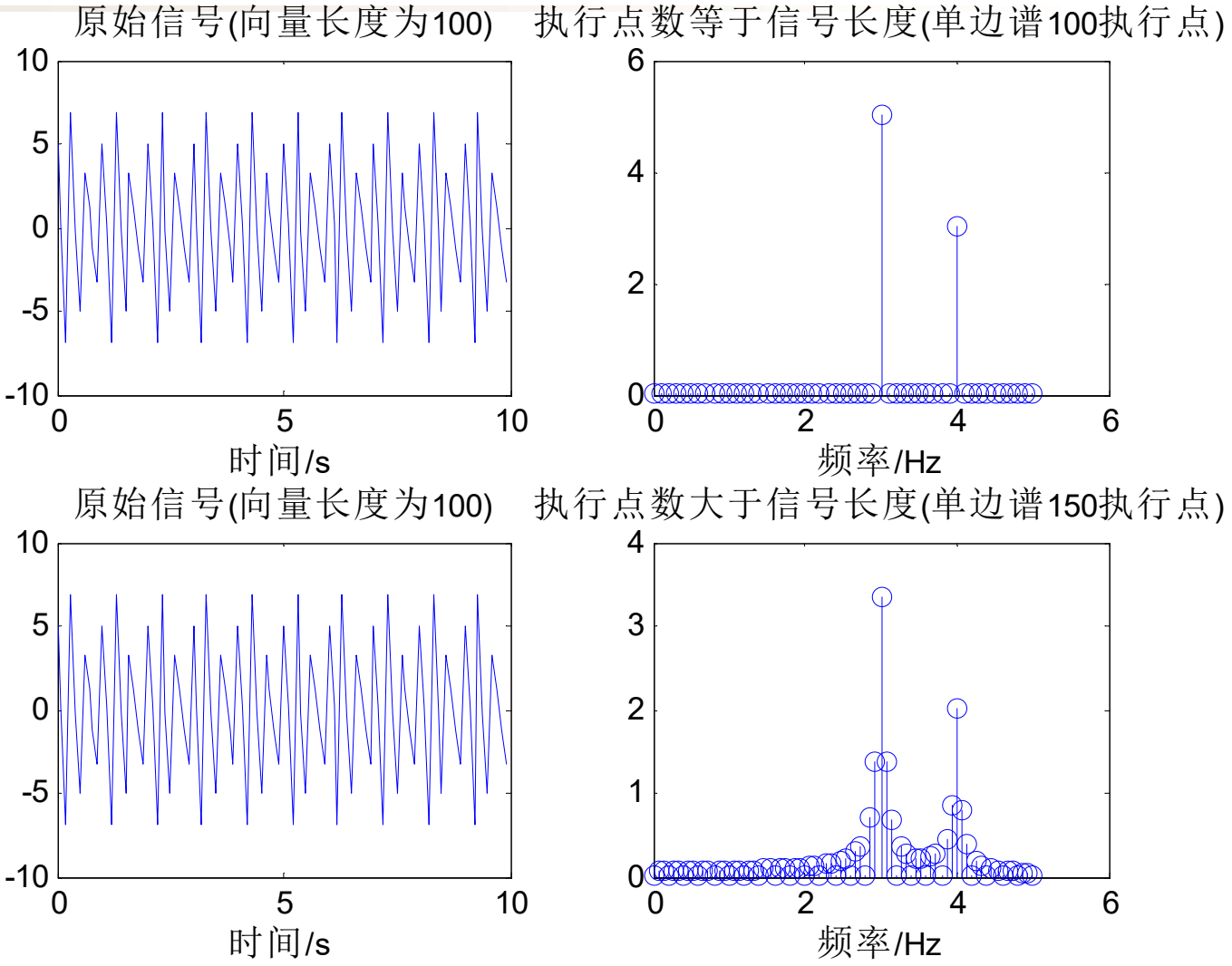
```
Fs=10;                          % 采样频率
dt=1/Fs;                        % 采样间隔
N=100;                          % 信号长度
t=[0:N-1]*dt;                   % 时间序列

xn=5*cos(6*pi*t)+3*sin(8*pi*t);
xn=[xn,zeros(1,N-100)];

subplot(2,2,1)
plot(t,xn)                      % 绘出原始信号
xlabel('时间/s'),title('原始信号(向量长度为100)')

% FFT分析 100点FFT
NN=N;
XN=fft(xn,NN);              % 共轭复数，具有对称性
f0=1/(dt*NN);                   % 基频
f=[0:ceil((NN-1)/2)]*f0;    % 频率序列
A=2*abs(XN)/NN;                     % 幅值序列
subplot(2,2,2),stem(f,A(1:ceil((NN-1)/2)+1)),xlabel('频率/Hz')   % 绘制频谱
title('执行点数等于信号长度(单边谱100执行点)');
```

将对N点FFT进行举例，说明当N大于向量y的长度时给频谱分析带来的变化。

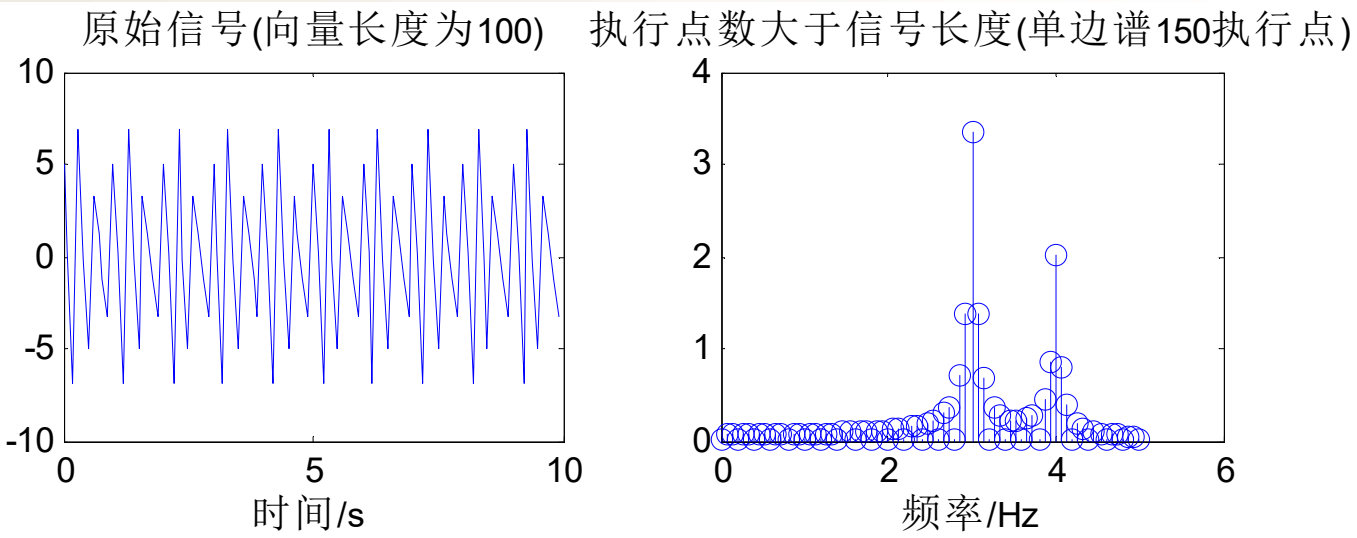
FFT 函数 例c0502.m

```
%% 执行FFT点数大于原信号长度
Fs=10;                          % 采样频率
dt=1/Fs;                        % 采样间隔
N=100;                          % 信号长度
t=[0:N-1]*dt;                   % 时间序列

xn=5*cos(6*pi*t)+3*sin(8*pi*t);
xn=[xn,zeros(1,N-100)];

subplot(2,2,3)
plot(t,xn)                      % 绘出原始信号
xlabel('时间/s'),title('原始信号(向量长度为100)')

% FFI分析,150点FFI

NN=150;  %
XN=fft(xn,NN);  % 共轭复数，具有对称性
f0=1/(dt*NN);  % 基频
f=[0:ceil((NN-1)/2)]*f0;  % 频率序列
A=2*abs(XN)/NN;  % 幅值序列
subplot(2,2,4),stem(f,A(1:ceil((NN-1)/2)+1)),xlabel('频率/Hz')  % 绘制频谱（变量）
title('执行点数大于信号长度(单边谱150执行点)');
```

FFT 函数 例c0502.m



FFT 函数 例c0502.m



左列为信号时域图形，右列为对应信号的频谱图。可以看出当N大于向量y的长度时，由于fft自动将信号值补零，所以频率发生了变化（增加多种频率的小振幅振动，主峰幅值被削弱）。

结论：使用N点FFT时，N大于向量的长度时将导致频谱失真。

FFT 函数 例c0502.m

stem函数

stem(Y) 将数据序列 Y 绘制为从沿 x 轴的基线延伸的针状图。各个数据值由终止每个针状图的圆指示。

如果 Y 是向量，x 轴的刻度范围是从 1 至 length(Y)。

如果 Y 是矩阵，则 stem 将根据相同的 x 值绘制行中的所有元素，并且 x 轴的刻度范围是从 1 至 Y 中的行数。

FFT 函数 例c0502.m

stem函数

stem(X,Y) 在 X 指定的值的位置绘制数据序列 Y。X 和 Y 输入必须是大小相同的向量或矩阵。另外，X 可以是行或列向量，Y 必须是包含 length(X) 行的矩阵。

如果 X 和 Y 都是向量，则 stem 将根据 X 中的对应项绘制 Y 中的各项。

如果 X 是向量，Y 是矩阵，则 stem 将根据 X 指定的值集绘制 Y 的每列，这样 Y 的一行中的所有元素都是根据相同的值而绘制。

如果 X 和 Y 都是矩阵，则 stem 将根据 X 的对应列绘制 Y 的列。

FFT 函数 例c0503.m

x=0.5*sin(2*pi*15*t)+2*sin(2*pi*40*t)， fs=100Hz, 绘制：

（1） 数据个数N=32， FFT所用的采样点数NFFT=32;

（2）N=32，NFFT=128;

（3）N=136，NFFT=128;

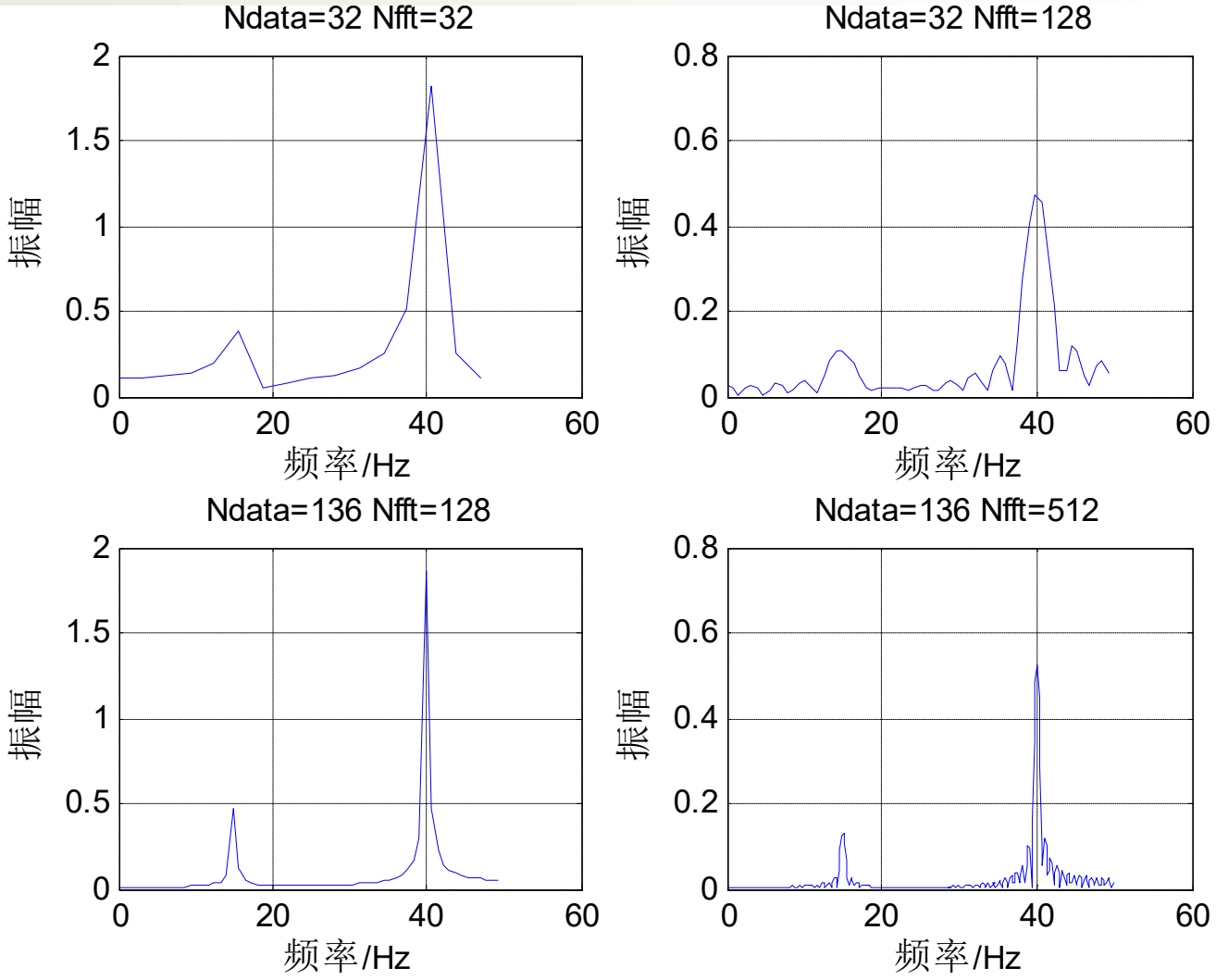
（4）N=136，NFFT=512。

FFT 函数 例c0503.m

```
fs=100; %采样频率
Ndata=32; %数据长度
N=32; %FFT的数据长度
n=0:Ndata-1;
t=n/fs;          %数据对应的时间序列

x=0.5*sin(2*pi*15*t)+2*sin(2*pi*40*t);    %时间域信号
y=fft(x,N);    %信号的Fourier变换
mag=abs(y);     %求取振幅
f=(0:N-1)*fs/N; %真实频率
subplot(2,2,1),plot(f(1:N/2),mag(1:N/2)*2/N); %绘出Nyqu
xlabel('频率/Hz');ylabel('振幅');
title('Ndata=32 Nfft=32');grid on;
```

```
Ndata=32;    %数据个数
N=128;       %FFT采用的数据长度

Ndata=136;    %数据个数
N=128;       %FFT采用的数据个数

Ndata=136;      %数据个数
N=512;      %FFT所用的数据个数
```

FFT 函数 例c0503.m



FFT 函数 例c0504.m

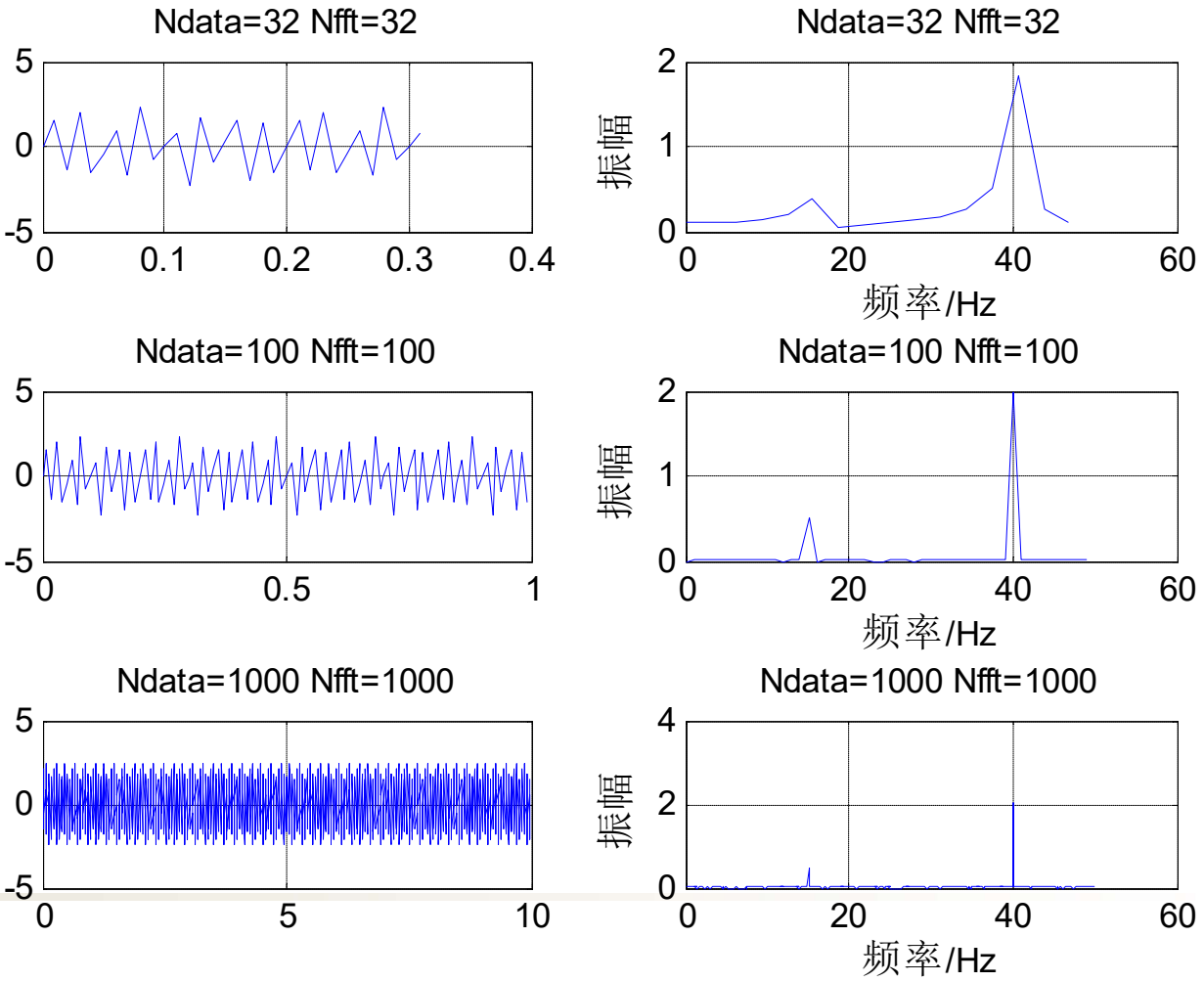
Fs不变，采样点数N改变，分辨率df改变

```
fs=100; %采样频率
Ndata=32; %数据长度
N=32; %FFI的数据长度
n=0:Ndata-1;
t=n/fs;          %数据对应的时间序列

x=0.5*sin(2*pi*15*t)+2*sin(2*pi*40*t);    %时间域

subplot(3,2,1),plot(t,x);
title('Ndata=32 Nfft=32');grid on;

y=fft(x,N);    %信号的Fourier变换
mag=abs(y);     %求取振幅
f=(0:N-1)*fs/N; %真实频率
subplot(3,2,2),plot(f(1:N/2),mag(1:N/2)*2/N); %
xlabel('频率/Hz');ylabel('振幅');
title('Ndata=32 Nfft=32');grid on;
```

```
Ndata=100; %数据长度
N=100; %FFI的数据长度

Ndata=1000; %数据长度
N=1000; %FFI的数据长度
```

FFT 函数 例c0504.m

Fs不变，采样点数N改变，分辨率df改变



fftshift函数

fftshift 将零频分量移到频谱中心 双边带的频谱

Y = fftshift(X) 通过将零频分量移动到数组中心，重新排列傅里叶变换 X。

如果 X 是向量，则 fftshift 会将 X 的左右两半部分进行交换。

如果 X 是矩阵，则 fftshift 会将 X 的第一象限与第三象限交换，将第二象限与第四象限交换。

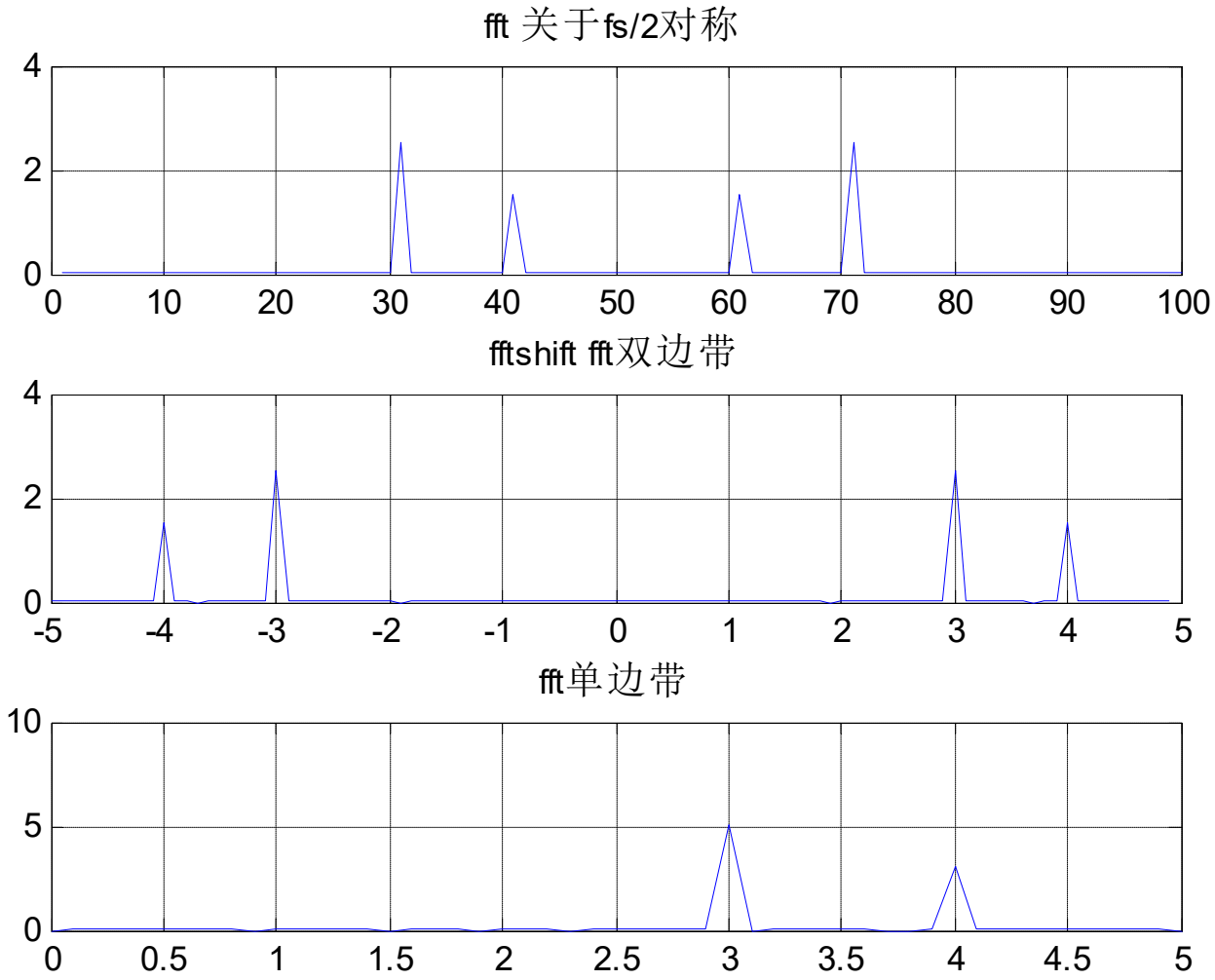
如果 X 是多维数组，则 fftshift 会沿每个维度交换 X 的半空间。

FFT 函数 例c0505.m

```
Fs = 10;                % 采样频率
T = 1/Fs;               % 采样间隔
L = 100;                % 信号长度
t = (0:L-1)*T;          % 时间序列

xt=5*cos(6*pi*t)+3*sin(8*pi*t);

Y1=fft(xt);
Y2=fftshift(Y1)
fshift = (-L/2:L/2-1)*(Fs/L);
powershift = abs(Y2)/L;

Y=abs(Y1/L);
P1 = Y(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);

f=Fs*(0:(L/2))/L;
```

```
subplot(311);plot(abs(Y1)/L);
title('fft 关于fs/2对称');grid on;
subplot(312);plot(fshift,powershift);
title('fftshift fft双边带');grid on;
subplot(313);plot(f,P1)
title('fft单边带');grid on;
```

FFT 函数 例c0505.m



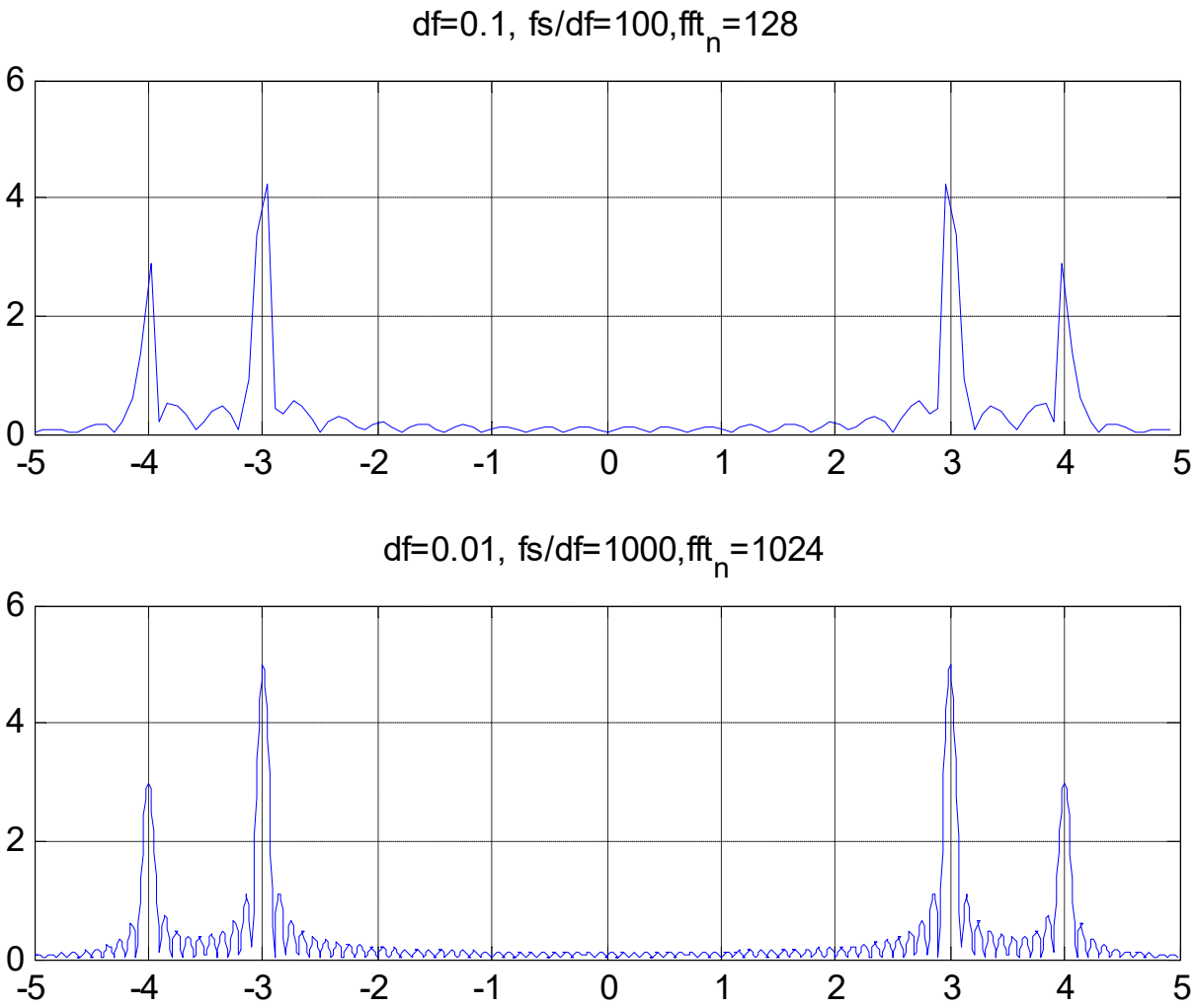
fftseq函数

```
function [M,m,df]=fftseq(m,ts,df)
fs=1/ts;
if nargin == 2
  n1=0;
else
  n1=fs/df;
end
n2=length(m);
n=2^(max(nextpow2(n1),nextpow2(n2)));
M=fft(m,n);
m=[m,zeros(1,n-n2)];
df=fs/n;
```

fftseq函数 例c0506.m

```
Fs = 10;                % 采样频率
T = 1/Fs;               % 采样间隔
L = 100;                % 信号长度
t = (0:L-1)*T;          % 时间序列
df=0.1                                          df=0.01

xt=5*cos(6*pi*t)+3*sin(8*pi*t);

[Y1,xr,df]=fftseq(xt,1/Fs,df) ;
NN=length(xr)

f= (-NN/2:NN/2-1)*df;
subplot(2,1,1),plot(f,fftshift(2*abs(Y1)/L));
title('df=0.1, fs/df=100,fft_n=128');grid on;
```

fftseq函数 例c0506.m



ifft函数

ifft 逆向快速傅里叶变换

X = ifft(Y) 使用快速傅里叶变换算法计算 Y 的逆离散傅里叶变换。X 与 Y 的大小相同。

如果 Y 是向量，则 ifft(Y) 返回该向量的逆变换。

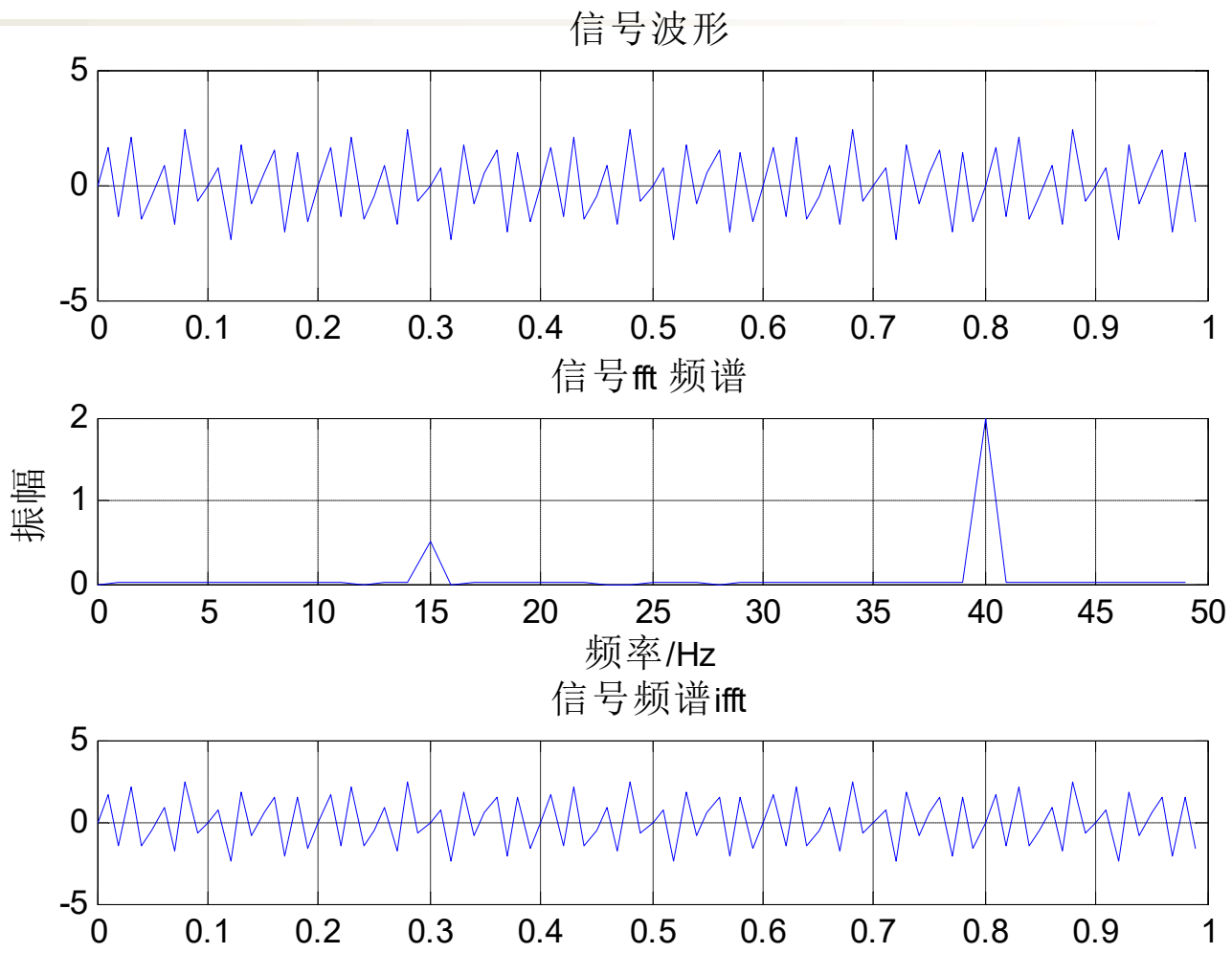
如果 Y 是矩阵，则 ifft(Y) 返回该矩阵每一列的逆变换。

如果 Y 是多维数组，则 ifft(Y) 将大小不等于 1 的第一个维度上的值视为向量，并返回每个向量的逆变换。

ifft函数 例c0507.m

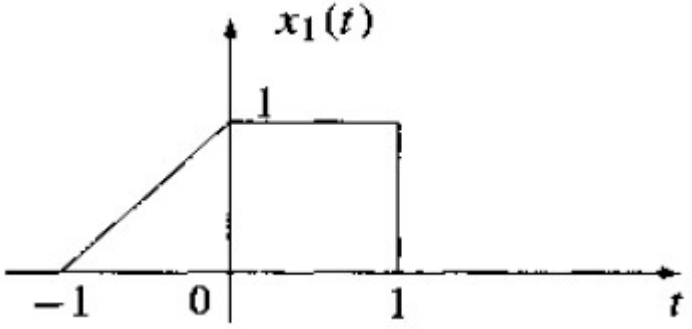
```
fs=100; %采样频率
Ndata=100; %数据长度
N=100; %FFI的数据长度
n=0:Ndata-1;
t=n/fs;          %数据对应的时间序列
x=0.5*sin(2*pi*15*t)+2*sin(2*pi*40*t);    %时间域信号
subplot(3,1,1),plot(t,x);
title('信号波形');grid on;

y=fft(x,N);    %信号的Fourier变换
mag=abs(y);     %求取振幅
f=(0:N-1)*fs/N; %真实频率
subplot(3,1,2),plot(f(1:N/2),mag(1:N/2)*2/N); %绘出Nyqui
xlabel('频率/Hz');ylabel('振幅');
title('信号fft 频谱');grid on;

xr=ifft(y)
subplot(3,1,3),plot(t,xr);
title('信号频谱ifft');grid on;
```

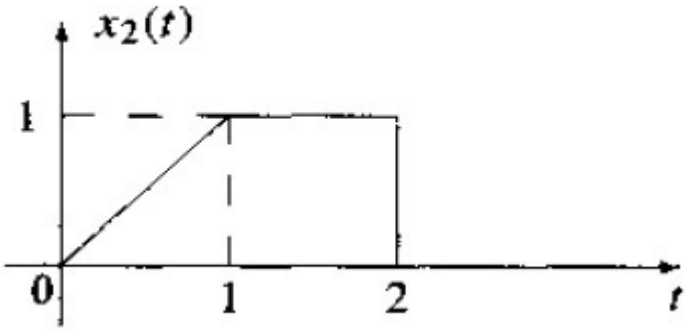
ifft函数 例c0507.m



例c0508.m

画出图中信号x1(t),x2(t)的幅度谱和相位谱。



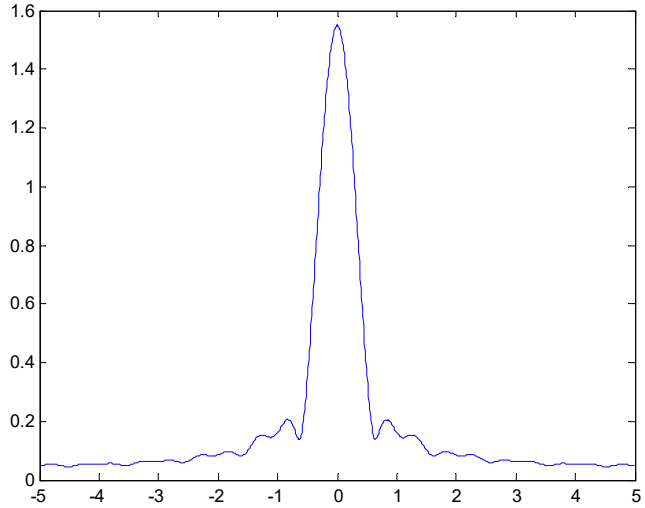


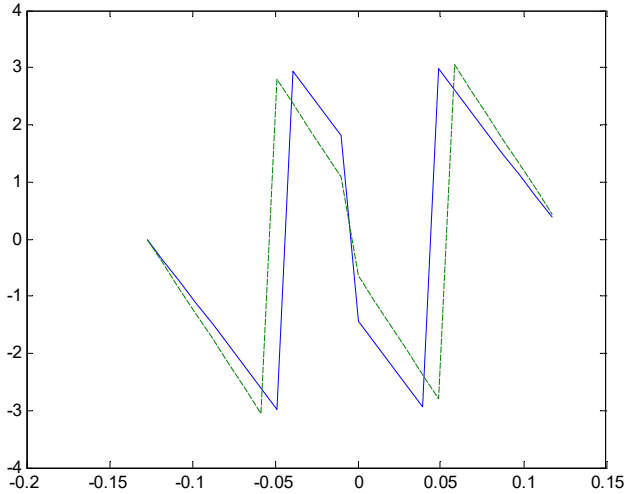
例c0508.m

```
df=0.01;
fs=10;
ts=1/fs;
t=[-5:ts:5];
x1=zeros(size(t));
x1(41:51)=t(41:51)+1;
x1(52:61)=ones(size(x1(52:61)));
x2=zeros(size(t));
x2(51:71)=x1(41:61);
[X1,x11,df1]=fftseq(x1,ts,df);
[X2,x21,df2]=fftseq(x2,ts,df);
X11=X1/fs;
X21=X2/fs;
f=[0:df1:df1*(length(x11)-1)]-fs/2;
figure(1);
plot(f,fftshift(abs(X11)))
figure(2);
plot(f(500:525),fftshift(angle(X11(500:525))),...
    f(500:525),fftshift(angle(X21(500:525))),'--')
```

例c0508.m

画出图中信号x1(t),x2(t)的幅度谱和相位谱。





除了时移以外，这两个信号是相同的，所以具有相同的幅度谱