2. 利用FFT函数完成傅立叶

	正变换	逆变换
CTFS	$T_s/T=1/N$	N
DTFS	$1/N$	N
CTFT	T_s	$N\Delta f=f_s=1/T_s$
DTFT	1	$1/T_s$

幅度调整

DFT

$$x[n]=\frac{1}{N}\sum_{k=0}^{N-1}X[k]W_N^{-nk},\quad n=\langle N\rangle$$

$$X[k]=\sum_{n=0}^{N-1}x[n]W_N^{nk},\quad k=\langle N\rangle$$

CTFS

$$x(t)=\sum_{k=-\infty}^{\infty}a_k\,\mathrm{e}^{jk\omega_0 t}$$

$$a_k=\frac{1}{T}\int_T x(t)\,\mathrm{e}^{-jk\omega_0 t}\,\mathrm{d}t$$

CTFT

$$x(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}X(j\omega)\,\mathrm{e}^{j\omega t}\,d\omega$$

$$X(j\omega)=\int_{-\infty}^{\infty}x(t)\,\mathrm{e}^{-j\omega t}\,\mathrm{d}t$$

可以把 DFT 看做一个计算模块。做其他变换要比较相关的公式进行振幅调整。

DTFS

$$x[n]=\sum_{k=\langle N\rangle}a_k\,\mathrm{e}^{jk\omega_0 n}$$

$$a_k=\frac{1}{N}\sum_{n=\langle N\rangle}x[n]\,\mathrm{e}^{-jk\omega_0 n}$$

DTFT

$$x[n]=\frac{1}{2\pi}\int_{2\pi}X\left(e^{j\omega}\right)e^{j\omega n}\,d\omega$$

$$X\left(e^{j\omega}\right)=\sum_{n=-\infty}^{\infty}x[n]e^{-j\omega n}$$

偶函数$u(t+T)-u(t-T)$

$$\mathcal{F}\{u(t+T)-u(t-T)\}=\frac{2\sin\omega T}{\omega}$$

脉冲宽度$w=2T, \omega=2\pi f, \operatorname{sinc}(x)=\dfrac{\sin(\pi x)}{\pi x}$

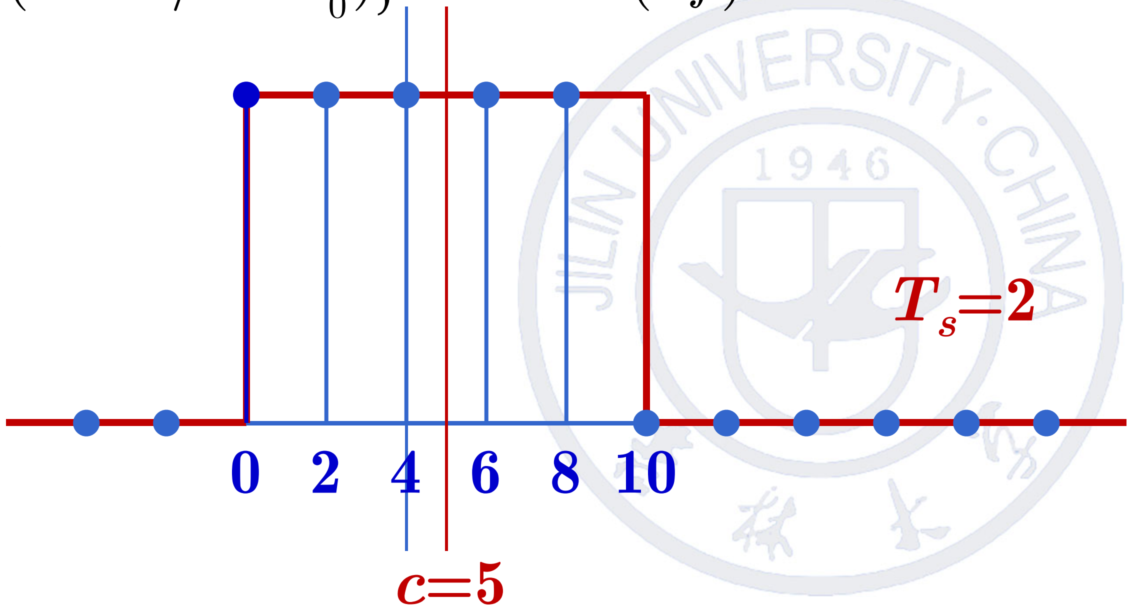
$$\mathcal{F}\{u(t+w/2)-u(t-w/2)\}=\frac{2\sin\pi w}{2\pi f}=w\operatorname{sinc}(wf)$$

若中心在t_0,

$$\mathcal{F}\{u(t+w/2-t_0)-u(t-w/2-t_0)\}=w\operatorname{sinc}(wf)e^{-j\omega t_0}$$

连续时间：$t_0=c$

离散时间：$t_0=c-\dfrac{T_s}{2}$



实践 ▸ 程序存档 ▸ 2 信号频谱分析

变量 - Z1

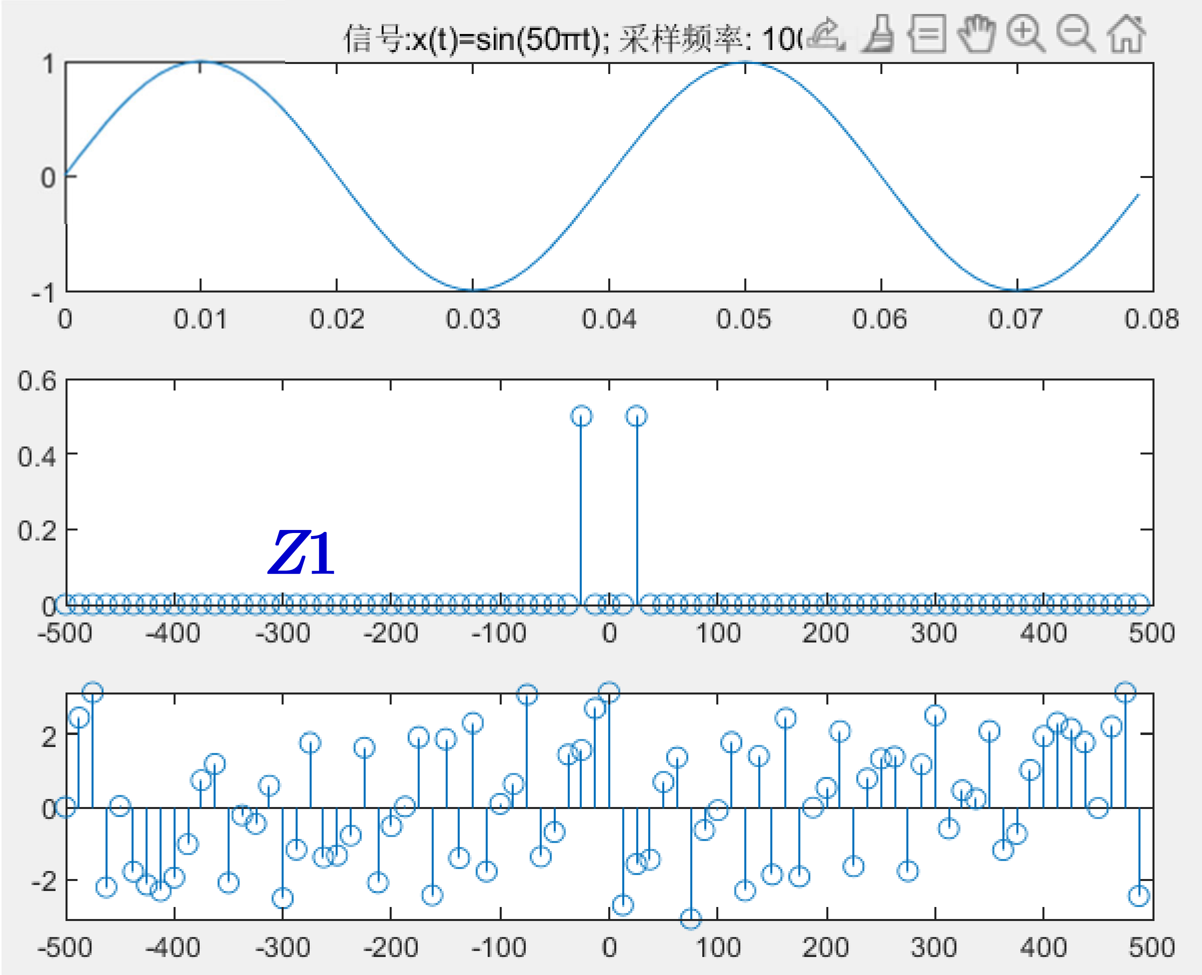
Z1

1x80 double

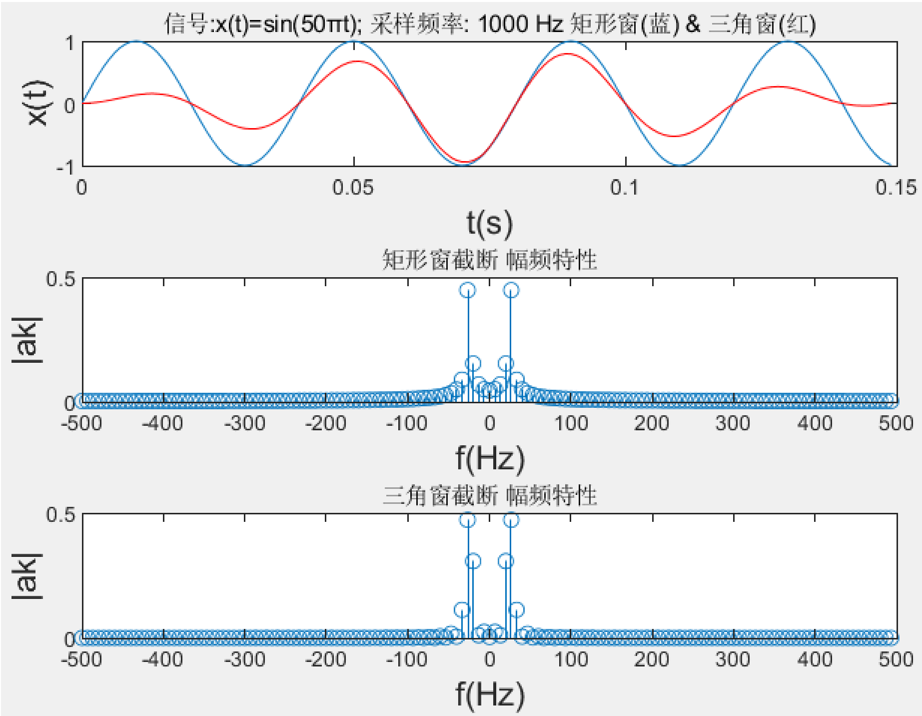
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	3.2327e-17	3.1874e-17	0.5000	5.0505e-17	5.0647e-17	2.5029e-17	4.7579e-17	3.6121e-17	3.5160e-17	4.2385e-17	2.6435e-17	4.5994e-17	4.4739e-17	4.9531e-17	3.2466e-17	3.4847e-17
2																

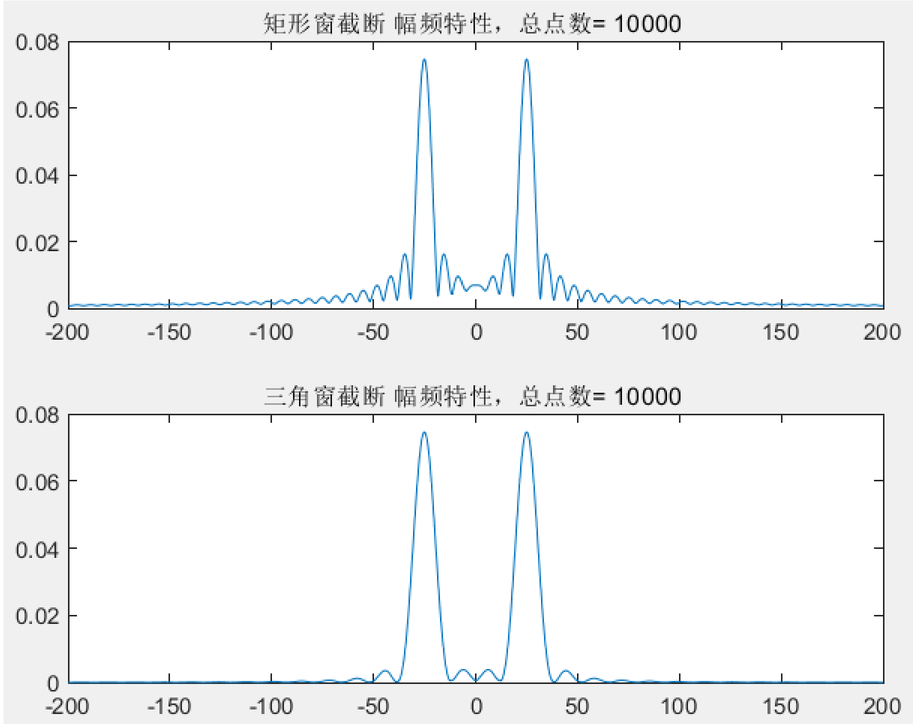
```
Z1=abs(y1); % 振幅谱
Z2=angle(y1); % 相位谱
for i=1:N1     % 微小信号的相位设置为零
    if Z1(i)<1E-10
        Z2(i)=0;
    end
end
```

或 `Z2=(sign(Z1-1E-10)+1)/2.*angle(y1); % 相位谱`



（a）利用FFT分析连续正弦信号的频谱：改变时间截断长度研究其对频谱泄漏的影响，**并研究采用不同窗函数对信号进行时域截断时，对频谱泄漏的影响**

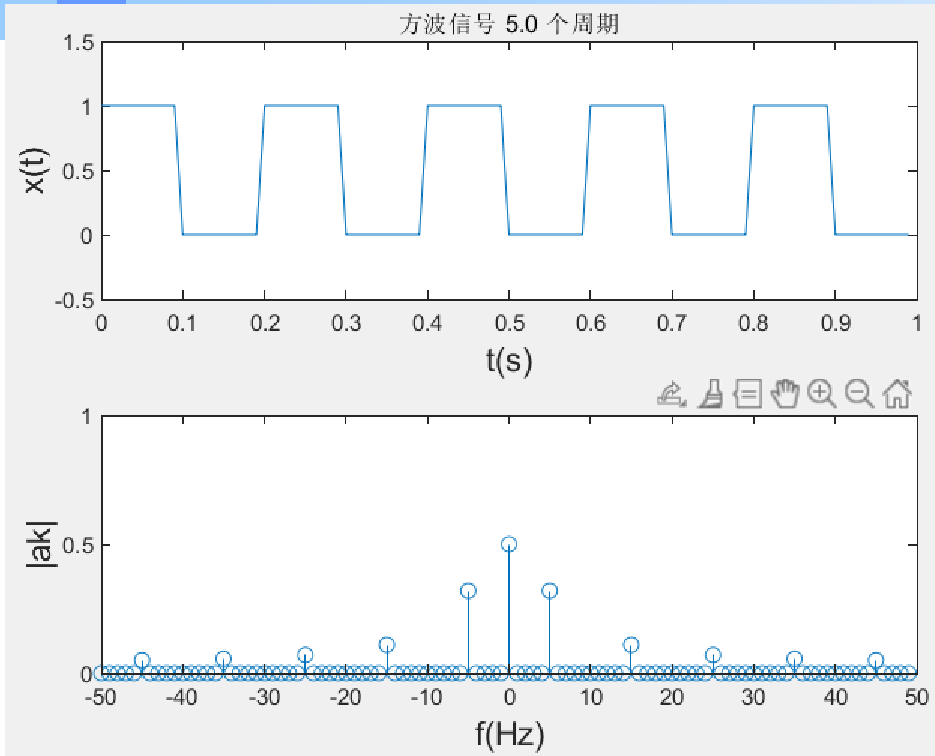


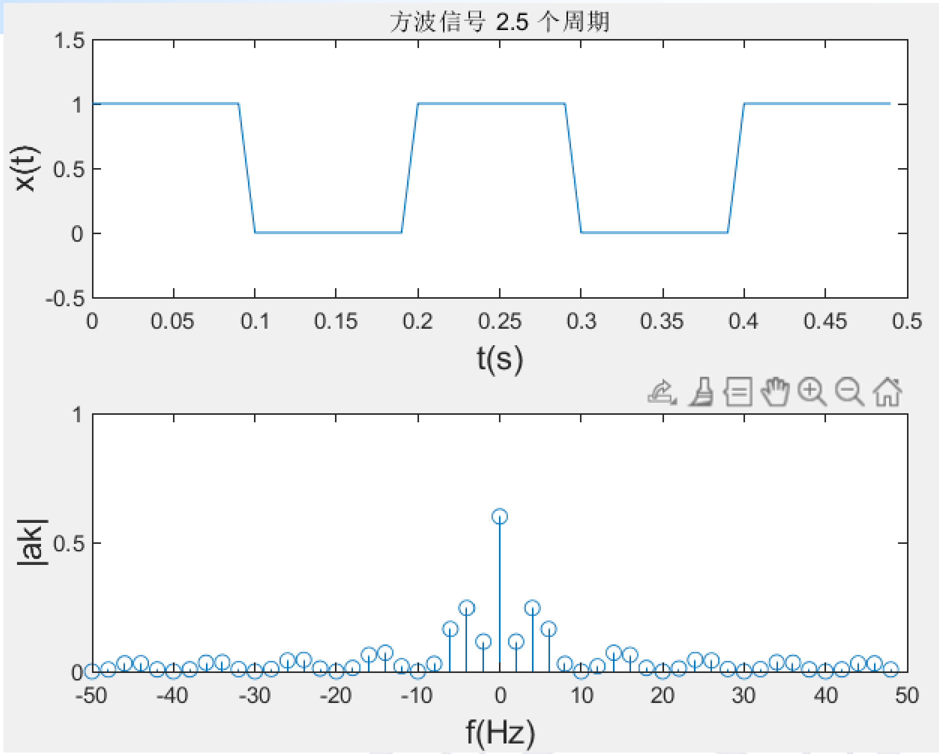



```
Freq=25; % Hz, 信号频率
fs=1000; % 采样频率，用于模拟连续信号，采样频率要很高：远高于信号频率
Ts=1/fs;
TT=0.15; % Total Time, 总时长
t=0:Ts:(TT-Ts);
N=length(t);
x1=sin(2*pi*Freq*t);
x2=x1.*bartlett(N)'; % 三角窗
y1=fft(x1)/N; % 未加窗
y2=fft(x2)/N;  % 加三角窗
Z1=abs(y1); % 未加窗幅度谱
Z2=abs(y2); % 加三角窗幅度谱
k=-N/2:N/2-1; % [-fs/2,fs/2] 的序号，假定N是偶数
f=k*fs/N;  % [-fs/2,fs/2]
```

```
M=10000;
y3=fft(x1,M)*Ts; %fft(x1,M) 自动完成补零
y4=fft(x2,M)*Ts; %fft(x2,M) 自动完成补零
f2=(-M/2:(M/2-1))*fs/M;
figure(2), subplot(2,1,1)
plot(f2,fftshift(abs(y3))); %y3: 未加窗，y4: 加窗
title(sprintf('矩形窗截断 幅频特性，总点数= %0.0f ',M))
xlim([-200,200]); % 只显示此频率范围
subplot(2,1,2)
plot(f2,2*fftshift(abs(y4)));%乘以2是因为加三角窗使信号能量减半（帕斯瓦尔定理）
                                % 为了和矩形窗作对比所以乘以2
title(sprintf('三角窗截断 幅频特性，总点数= %0.0f ',M))
xlim([-200,200]); % 只显示此频率范围
```

（b）利用FFT，分析连续周期方波信号的频谱：通过改变采样率与时间截断长度,分别研究频谱的混叠与泄漏

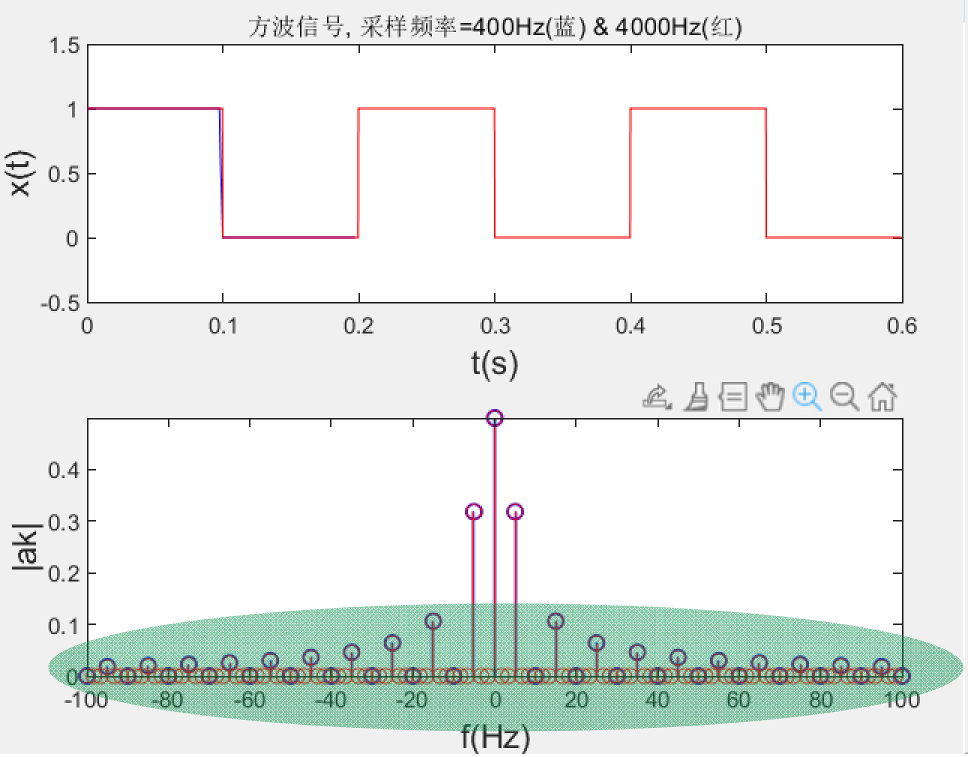


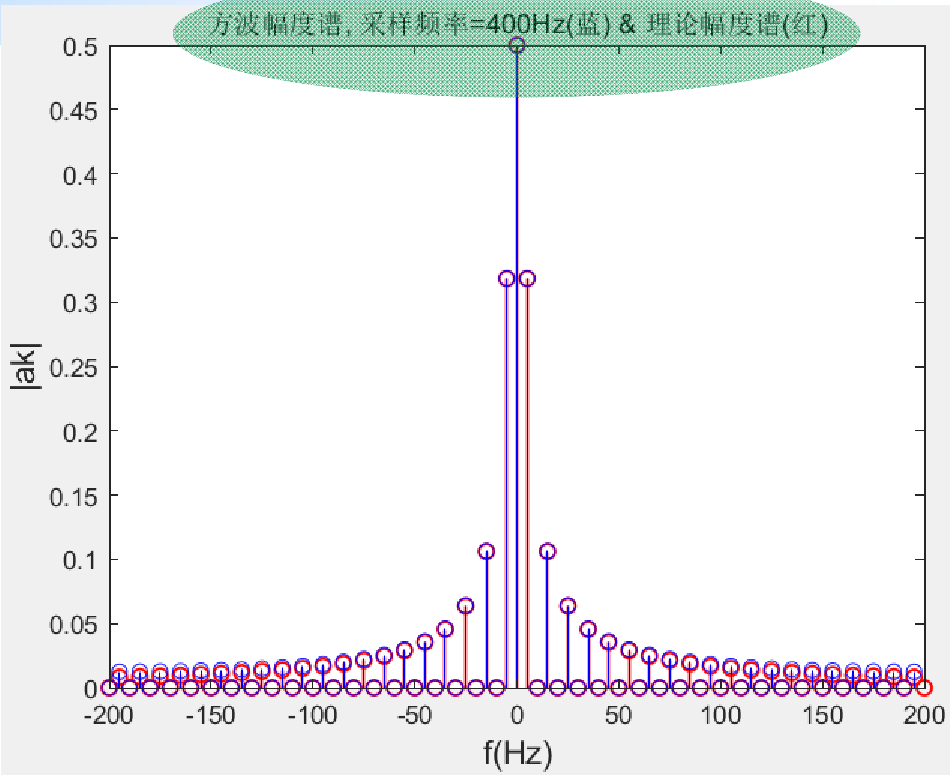



```
fs=100; %采样频率
ts=1/fs; % 采样间隔
Freq=5;
%截取整周期，没有发生频谱泄露
TT=1;   % 5个周期
t1=0:ts:TT-ts;  % 时间序列
y1=(square(2*pi*Freq*t1)+1)/2; % 周期方波
N=length(t1);
if mod(N,2)==0   % 频率序列，  依N 的奇偶
   f=(-N/2:N/2-1)*fs/N;
else
   f=(-(N-1)/2:(N-1)/2)*fs/N;
end
y2=fft(y1)/N; % DTFS  /N
Z1=abs(y2); % 幅度谱
```

```
figure(1)
subplot(2,1,1),plot(t1,y1);
ylim([-0.5,1.5]);
title(sprintf('方波信号 %0.1f 个周期',TT*Freq));
xlabel('t(s)','FontSize',13);
ylabel('x(t)','FontSize',13);
subplot(2,1,2),stem(f,fftshift(Z1));
axis([-50,50,0,1]);
xlabel('f(Hz)','FontSize',13);
ylabel('|ak|','FontSize',13);
```

（b）利用FFT，分析连续周期方波信号的频谱：通过改变采样率与时间截断长度,分别研究频谱的混叠与泄漏





对比不同采样率对方波频谱混叠的影响，低采样率混叠更明显，表现为幅频特性略高。对于方波这种非带限信号 (含有无限高的频率成分)，只要是通过采样，之后应用fft函数获得频谱曲线，都会发生混叠，采样率越低频谱混叠越严重。

```
fs1=400;  % 采样频率1
fs2=4000; % 采样频率2
TT1=0.2; % 总时长
TT2=0.6;
Freq=5; % 信号频率， Hz
ts1=1/fs1; % 采样间隔
ts2=1/fs2;
t1=0:ts1:TT1-ts1;
x1=(square(2*pi*Freq*t1,50)+1)/2;  % 50% 占空比
t2=0:ts2:TT2-ts2;
x2=(square(2*pi*Freq*t2,50)+1)/2;
N=length(t1);
M=length(t2);
y1=fft(x1)/N;    % 用DFT做DTFT计算，幅度调整 /N
y2=fft(x2)/M;
```

```
%对比不混叠情况与采样混叠情况的幅频特性，不混叠频谱是通过理论计算得到的。

k1=-40:-1;  % 频率范围：(-200Hz, 200Hz)
k2=1:40;
k=[k1,0,k2];
f3=k*Freq;

ak1=(sin(k1*pi/2)./k1/pi).*exp(-1i*k1*2*pi*5*0.04875);% (0.1-1/400)/2 = 0.04875
ak2=(sin(k2*pi/2)./k2/pi).*exp(-1i*k2*2*pi*5*0.04875);

ak=[ak1,0.5,ak2]; % 求sinc() 但避免除零
akabs=abs(ak);
```

画相位谱时才用得着