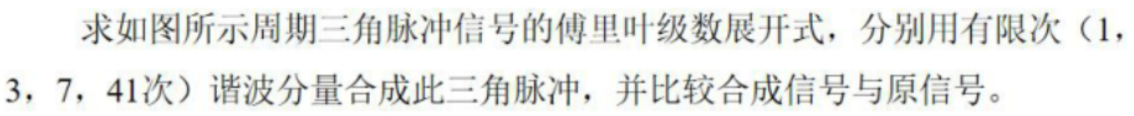
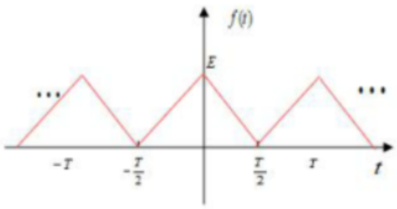
实验三

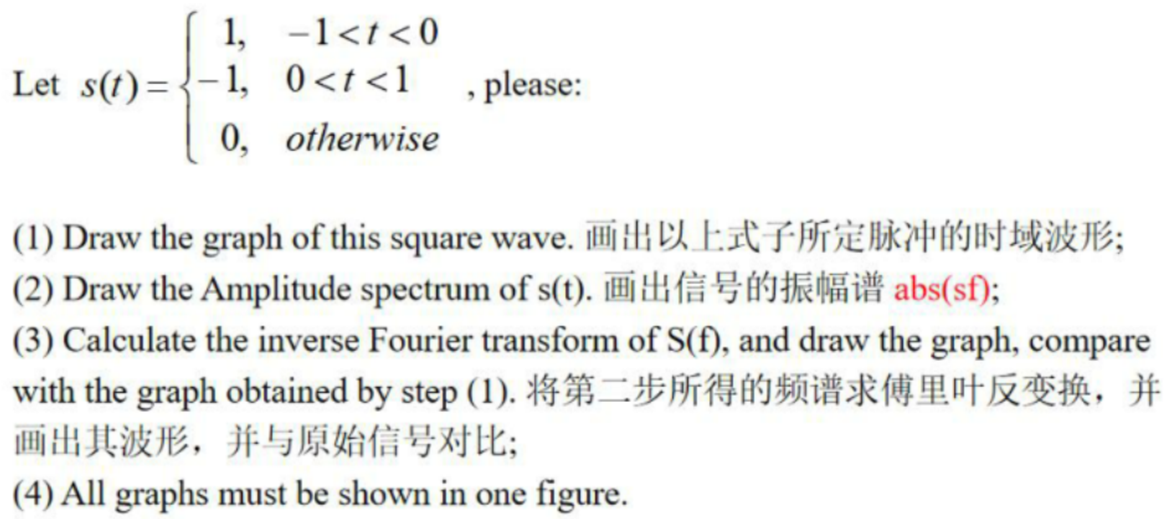


求如图所示周期三角脉冲信号的傅里叶级数展开式，分别用有限次（1，3，7，41次）谐波分量合成此三角脉冲，并比较合成信号与原信号。



代码:

```
%定义参数
E = 1;%三角波幅度
T = 2;%三角波周期
%定义时间范围
t = linspace(-3*T,3*T, 1000);
%定义三角波信号
f_t = E+ E * 2 * abs(mod(t, T) - T/2) / T;
另画出原始三角波信号
figures
subplot(3,2,1);
plot(t,ft);
title('原始三角波信号");
%定义傅里叶级数展开式
Nvalues = [1, 3, 7, 41];%合成信号中谐波的数量
% 合成信号
for i = 1:length(N_values)
    N = Nvalues(i);
    f_reconstructed = zeros(size(t));
end
for n = 1:2:N
    harmonic = (4 * E / (pi^2)) * (1 / n^2) * cos(2 * pi * n * t / T);
    f_reconstructed = f_reconstructed + harmonic;
end
%画出合成信号
subplot(3, 2,i + 1);
plot(t, f_reconstructed);
title(['合成信号，谐波数量：", numzstr(N)]);
End
```

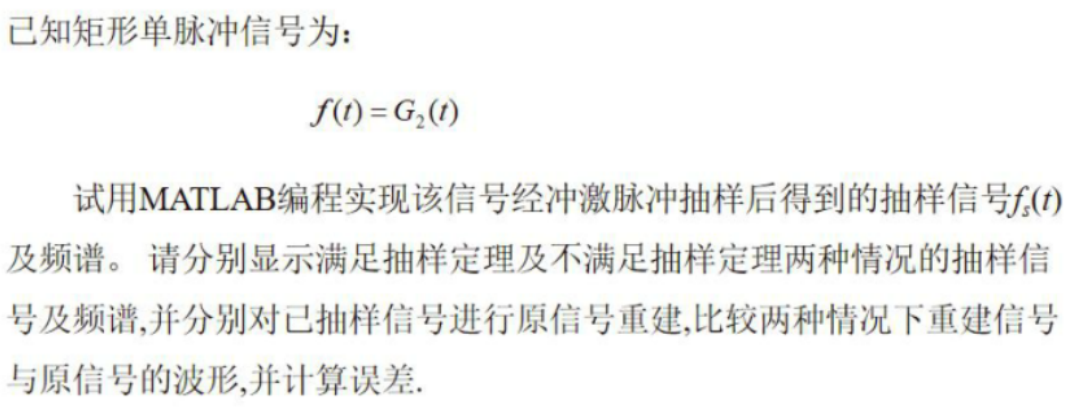


Let $s(t)=\begin{cases}1, & -1<t<0\\ -1, & 0<t<1\\ 0, & otherwise\end{cases}$, please:

(1) Draw the graph of this square wave. 画出以上式子所定脉冲的时域波形;
(2) Draw the Amplitude spectrum of s(t). 画出信号的振幅谱 abs(sf);
(3) Calculate the inverse Fourier transform of S(f), and draw the graph, compare with the graph obtained by step (1). 将第二步所得的频谱求傅里叶反变换，并画出其波形，并与原始信号对比;
(4) All graphs must be shown in one figure.

代码:

```
%定义信号 s(t)
t = linspace(-2,2, 1000);
s_t = zeros(size(t));
s_t(t ⩾ 0 &t < 1) = -1;
s t(t ⩾ -1 &t <0) = 1j;
%画出时域波形
figure;
subplot(2,2,1);
plot(t,s_t);
ttle('时域波形');
%计算信号的傅里叶变换
S_f= fftshift(fft(s_t));
%画出振幅谱
frequencies = 1inspace(-0.5, 0.5, length(S_f));
subplot(2,2,2);
plot(frequencies, abs(S_f));
title("振幅谱')
%计算傅里叶反变换
s_t_reconstructed = ifft(ifftshift(S_f));
%画出傅里叶反变换后的信号波形，并与原始信号对比
subplot(2,2, [3,4]);
plot(t, s_t, 'b', t, real(s_t_reconstructed), 'r--');
legend("原始信号", "重建信号");
title("原始信号与重建信号对比");
%调整图像显示格式
xlim([-2,2]);
```



已知矩形单脉冲信号为：

$$f(t)=G_2(t)$$

试用MATLAB编程实现该信号经冲激脉冲抽样后得到的抽样信号$f_s(t)$及频谱。 请分别显示满足抽样定理及不满足抽样定理两种情况的抽样信号及频谱,并分别对已抽样信号进行原信号重建,比较两种情况下重建信号与原信号的波形,并计算误差.

```
% 定义矩形单脉冲信号
t= linspace(-5,5,1000);
ft=rectpuls(t);
%抽样定理满足的情况
Fs_salisDy=20:%满足抽样定理的采样频率
TsLsatisfy=1/Fs_satisty:
% 构造单位冲激序列
nLsatisDy=-20:20;
delta_n_satisDy= Ts_satisfy * (n_salisfy==0);
%信号经过抽样
fs_t_satis[y= conv(ft, delta_n_satisfy):
is_tsatisDy= Is_tLsatisiy(1:length()
%计算频谱
frequencies-satisfy=linspace(-Fs-satisfy/2, Fs_satis[y/2, length(fs_t_satis[y));
fs_f_satisDy= ffshift(ff(fs_tLsatisfy)):
% 原信号重建
fLtLreconstructLsatisfy= interpl(t. fsLt_satisfy.t *FsLsatisfv, 'spline');
% 计算误差
error_satisDy = norm(ft - f_t_reconstruct_satis[y) /norm(ft)
% 不满足抽样定理的情况
FsLunsatisfty= 10
TsLunsatisfy=1/Fs_unsatisty:
%构造单位冲激序列
nLunsatisfy= -10:10
delta_n-unsatisfy = Ts_unsatisDy * (n_unsatisfy ==0):
%信号经过抽样
fs_t_unsatis[y = conv(ft, delta_n_unsatisfy);
isLtLunsatisDy= fsLLunsatisDll:length(t):% 裁剪长度与原始信号相同
% 计算频谱
frequenciesLunsatisiy=linspace(-
Fs_unsatisfy/2,F'sLunsatisfy/2,length(fs_t_unsatisfy));
fs_funsatisty = fitshif(ft(isLtLunsatisD):
% 原信号重建
f_t_reconstruct_unsatis[y = interpl(, fs_tunsalis[y, t * FsLunsatisDy,
'spline');
%计算误差
error_unsatisty= norm(ft-f_t_reconstruct_unsatisfy) /norm(ft):
% 绘图展示结果
figure;
subplot(7,1,1):
plot(t,f):
```

```
title(原始信号");
subplot(7,1,2):
plot(frequencies_satisfy, abs(fs_f_satisfy));
tle(频谱图(满足抽样定理);
subplot(7,1,3):
plot(t, fs_tLsatisfy):
tiue(抽样信号 （满足抽样定理)):
subplot(7,1,4);
plot(t,f_t_reconstructLsatisfy);
tide(重建信号 （满足抽样定理));
subplot(7,1,5):
plot(frequenciesLunsatisDy, abs(fs_f_unsatisDy);
title("频谱图(不满足抽样定理));
subplot(7,1,6):
plot(t, fs_tLunsatisfy):
tite(抽样信号(不满足抽样定理);
subplot(7,1,7):
plot(t, f_t_reconstruct_unsatisfy);
title("重建信号 （不满足抽样定理);
fprinti("重构误差 （满足抽样定理):%.4in',errorLsatisM);
fprint(重构误差 （不满足抽样定理):%.4n',enrorLunsatisM);
```