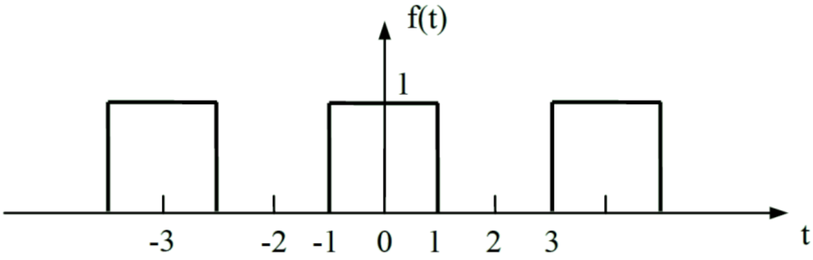
信号与系统实验报告 实验三

2019 年 11 月 9 日

1 题目一

1.1 题目内容

求下图所示周期矩形脉冲的傅里叶级数表达式，并用 MATLAB 求出前 3、7、15、30 次谐波合成的信号近似波形。信号采样区间请合理确定，绘图要标注清晰。



1.2 计算过程

该信号的基波周期$T_0 = 4$，则$\omega_0 = \frac{2\pi}{T_0} = \frac{\pi}{2}$，设信号$x(t)$的傅里叶级数表达式为：

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

求解得：

$$a_k = \frac{1}{T_0}\int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} x(t)e^{-jk\omega_0 t}\mathrm{d}t = \frac{1}{4}\int_{-1}^{1} e^{-jk\omega_0 t}\mathrm{d}t = \frac{\sin(k\omega_0)}{2k\omega_0} = \frac{1}{2}\frac{\sin(k\pi/2)}{k\pi/2}$$

由前N次谐波合成的信号近似波形为：

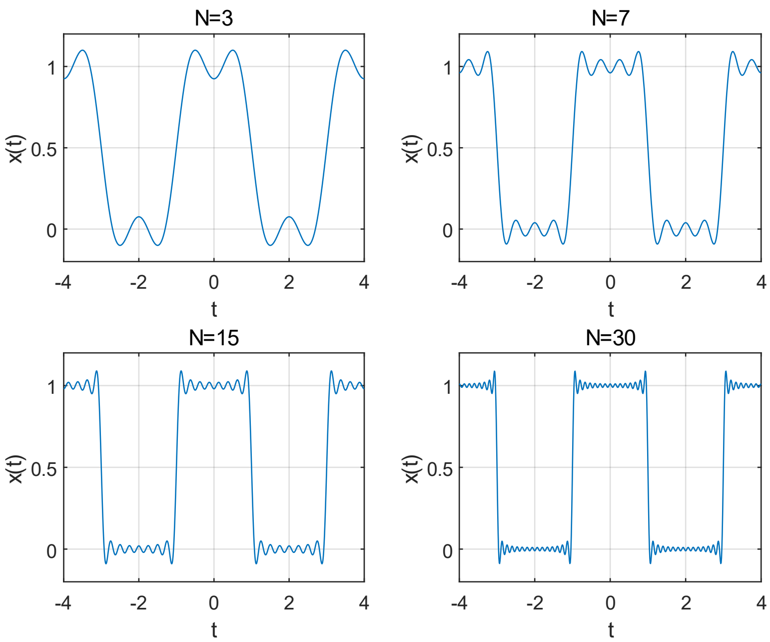
$$f_N(t) = \sum_{k=-N}^{N} a_k e^{jk\omega_o t} = \frac{1}{2} + \sum_{k=1}^{N} \frac{\sin(k\pi/2)}{k\pi/2}\cos\left(\frac{k\pi t}{2}\right)$$

1.3 实现代码

```
clc;
```

```
clear;
t = -4:0.01:4;
c0 = 0.5;
ns = [3, 7, 15, 30];
for ii=1:4
    subplot(2, 2, ii);
    n = ns(ii);
    fn = c0 * ones(1, length(t));
    for k=1:2:n
        fn = fn + cos(pi/2*k*t)*sin(k*pi/2)/(k*pi/2);
    end
    plot(t, fn);
    grid;
    title(['N=', num2str(n)]);
    xlabel('t');
    ylabel('x(t)');
    axis([-4, 4, -0.2, 1.2]);
end
```

1.4 运行结果



2 题目二

2.1 题目内容

产生并绘制周期为π，幅度为2的方波信号。分别绘制原始方波与 1 次、1 次和 3 次、1 次 3 次和 5 次、1 次 3 次 5 次和 7 次谐波相加的图像，各个谐波叠加的过程要求使用 for 循环实现。绘图区间至少包括两个完整的方波周期并有完整标注。取傅里叶级数的过程应体现在报告中。

2.2 方法对比

若要得到周期为π，幅度为2的方波信号有两种方式：第一种是直接构造一个方波信号，将定义域分为等长的区间，然后给出相应的值，也就是教程中的方法。第二种是利用 square 函数产生方波。在产生信号的过程中，我注意到两种方法得到的结果是有差异的，选择较小步长时差别较小，但是步长较大时会有明显差别，为了进一步探究这个问题，我选择不同的步长用上述两种方法绘制了方波信号。通过几幅图之间的对比，我们可以发现用第一种方法构造出的方波信号在步长较大时会有明显的误差，而 square 函数仍能保持较准确的结果。这可能是由于计算等长区间时，由于对小数进行了四舍五入，从而导致了精度的损失，致使构造得到的信号也会与真实信号有所偏差。并且使用 square 函数能够更加广泛地用于各种场景，在定义域改变时也能很容易适应，所以最终我采用了 square 函数的方法来构造方波信号。

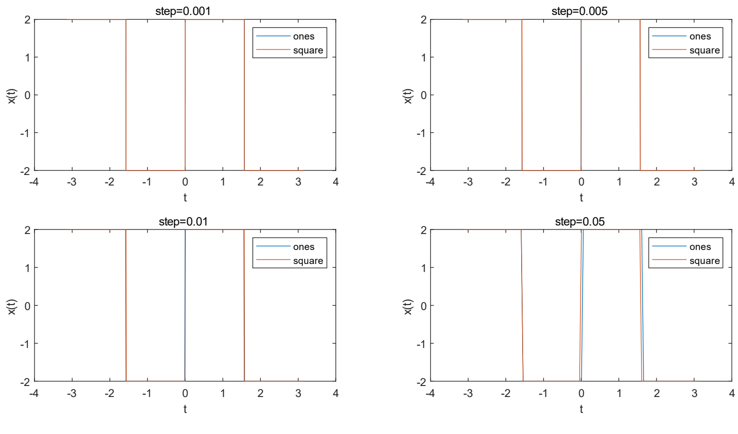


图 1 两种方法在不同步长下产生的方波信号

2.3 计算过程

该信号的基波周期为$T_0=\pi$，则$\omega_0=\frac{2\pi}{T_0}=2$，设信号$x(t)$的傅里叶级数表达式为：

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

求解得：

$$a_k=\frac{1}{T_0}\int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} x(t)e^{-jk\omega_0 t}\mathrm{d}t=\frac{2}{\pi}\left(\int_{-\frac{\pi}{2}}^{0} -e^{-jk\omega_0 t}\mathrm{d}t+\int_{0}^{\frac{\pi}{2}} e^{-jk\omega_0 t}\mathrm{d}t\right)=\frac{2}{\pi}\frac{1-(-1)^k}{jk}$$

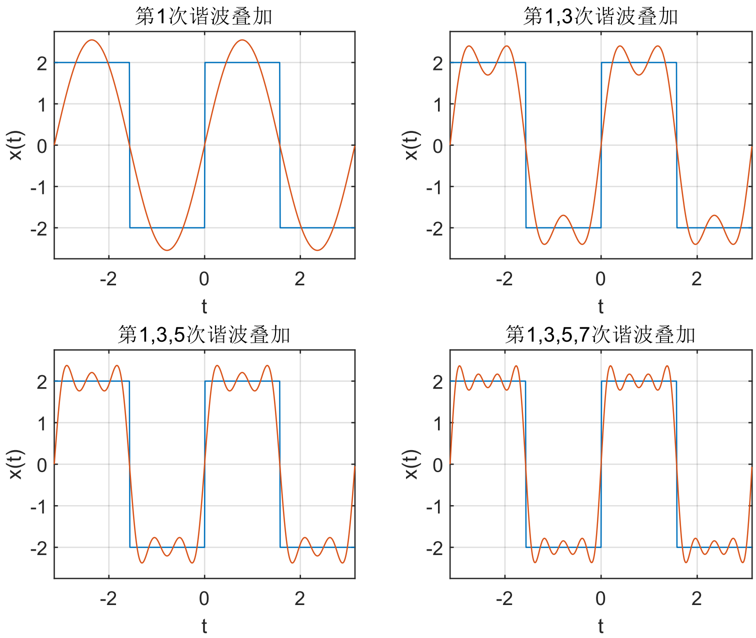
由前N次谐波合成的信号近似波形为：

$$f_N(t)=\sum_{k=-N}^{N} a_k e^{jk\omega_0 t}=\sum_{r=1}^{\lceil N/2\rceil}\frac{8}{(2r-1)\pi}\sin[2(2r-1)t]$$

2.4 实现代码

```
clc; clear;
t = -pi:0.01:pi;
n = floor(length(t)/4);
f = 2 * square(2*t, 50);
for n=1:4
    subplot(2, 2, n);
    fn = zeros(1, length(t));
    kk = '1';
    for k=1:2:2*n-1
        fn = fn + 8/pi/k*sin(2*k*t);
        if k>1
            kk = strcat(kk, ',', num2str(k));
        end
    end
    plot(t, f);
    hold on;
    plot(t, fn);
    grid;
    title(strcat('第', kk, '次谐波叠加'));
    xlabel('t');
    ylabel('x(t)');
    axis([-pi, pi, -2.75, 2.75]);
end
```

2.5 运行结果



3 题目三

3.1 题目内容

绘制周期三角波，其在一个周期内的表达式为：

$$f(t)=\begin{cases}2t & |t|\le\frac{1}{2}\\ 2(1-t) & \frac{1}{2}<t\le\frac{3}{2}\end{cases}$$

求其傅里叶级数，并绘制幅度、相位谱。求取傅里叶级数的过程应体现在报告中，绘制频谱图可调用函数，详见 pdf。

3.2 计算过程

该信号的基波周期$T_0=2$，则$\omega_0=\frac{2\pi}{T_0}=\pi$，设信号$x(t)$的傅里叶级数表达式为：

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

求解得：

$$a_k = \frac{1}{2}\int_{-1/2}^{3/2} x(t)e^{-jk\omega_0 t}\mathrm{d}t = \frac{1}{2}\left(\int_{-1/2}^{3/2} 2te^{-jk\pi t}\mathrm{d}t - \int_{1/2}^{3/2} 2e^{-jk\pi t}\mathrm{d}t\right)$$

化简得：

$$a_k = \begin{cases} \frac{-4j}{k^2\pi^2}\sin\left(\frac{k\pi}{2}\right) & k \neq 0 \\ 0 & k = 0 \end{cases}$$

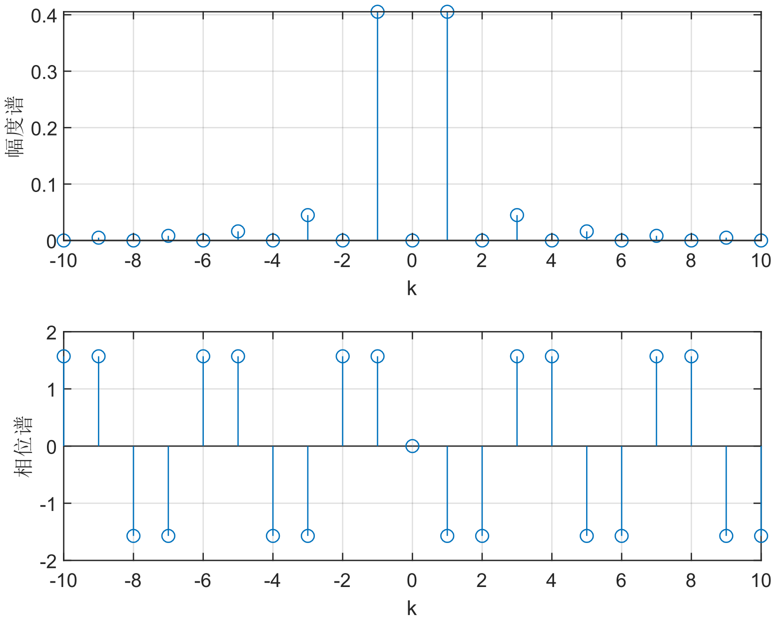
由前N次谐波合成的信号近似波形为：

$$f_N(t) = \sum_{k=-N}^{N} a_k e^{jk\omega_0 t} = \sum_{k=1}^{N} \frac{8}{k^2\pi^2}\sin\left(\frac{k\pi}{2}\right)\sin(k\pi t)$$

3.3 实现代码

```
clc;
clear;
N = 10;
n1 = -N:-1;
C1 = -4*1j/pi^2./n1.^2.*sin(n1*pi/2);
C0 = 0;
n2 = 1:N;
C2 = -4*1j/pi.^2./n2.^2.*sin(n2*pi/2);
Cn = [C1, C0, C2];
n = -N:N;
subplot(2, 1, 1);
stem(n, abs(Cn));
xlabel('k');
ylabel('幅度谱');
subplot(2, 1, 2);
stem(n, angle(Cn));
xlabel('k');
ylabel('相位谱');
```

3.4 运行结果



4 思考题

用构造波形的方法：

优点：简单直观，并且不需要变换。

缺点：定义域改变后需要重新计算；波形会受四舍五入精度的影响。

用 square 函数的方法：

优点：定义域改变后无需重新计算；绘制出来的波形较为准确。

缺点：需要将原始信号经过变换才能得到目标波形。

5 实验收获与感想

本次实验的核心是傅里叶级数的求解，并且用谐波来近似原始波形。通过本次实验，我对傅里叶级数有了进一步的理解。在实验过程中，比较复杂的部分就是傅里叶级数的计算，在计算之后我们可以观察近似的波形来验证计算结果是否正确。实验中我们还对两种绘制方波的方法进行了对比，探讨了这两种方法各自的优劣，也使我对 MATLAB 编程中实现的方法有了更深刻的理解。