

西安电子科技大学网络与信息安全学院
信号与系统实验报告

班 级： _

学 号： _

姓 名： _

电子邮箱：

指导教师： _

2024 年 5 月 6 日

实验题目：

1. 写出由程序

```
t=-2*pi:0.001:2*pi;
```

```
y=sawtooth(0.5*t,1);
```

```
plot(t,y)
```

形成的信号经周期延拓得到的周期信号的时域表达式($T=4\pi$)；编程计算其指数形式的傅里叶系数(计算至11次谐波)；用MATLAB画出前11次谐波叠加的波形，并指明吉布斯现象出现于何处。

实验摘要 (简单描述实验内容)：

写出周期信号的时域表达式，求前 **11** 次谐波的傅里叶级数，并画出图形、指明吉布斯现象

题目描述 (以中文简要描述实验题目, 建议陈述自己对题目的理解):

Sawtooth(0.5*t,1)。0.5*t 将时间向量 **t** 的值乘以 0.5, 这样可以改变波形的频率, 使得它的周期变为原来的 2 倍, 即原来的一半。第二个参数 **1** 指定了波形的幅度, 这意味着波形的振幅为 1。

由参考资料: 默认情况下, 周期是 2π , 则 **y** 的周期变为了 4π , 当 **t** 为 -2π 到 2π 时, 正好是一个周期。

MATLAB 中, **fft()** 函数提供了快速傅里叶变换, 但是得到的结果是信号的频域表示, 而不是傅里叶系数, **MATLAB** 并没有自带的求解傅里叶级数的函数

指数形式的傅里叶展开式为

$$f(t) = \sum_{n=-\infty}^{\infty} F_n e^{jn\Omega t}$$

实验要求11次谐波叠加

周期是 4π , 故 $\omega = \frac{1}{2}$

本题中, $f(t) = \sum_{n=-11}^{11} F_n e^{\frac{1}{2}jnt}$

我打算用自己积分的方式, 算出傅里叶系数, 然后叠加

实验过程与分析 (详细实验内容):

自己积分, 编写 **MATLAB** 代码:

```

T = 4*pi;
t = -2*pi:0.001:2*pi;
y = sawtooth(0.5*t, 1);

N = 11; % 谐波次数
ns = -N:N;

Fn = zeros(size(ns)); % 傅里叶级数向量, 对应 n 的值分别是 -11, -10 ... 0, 1 ... 11
for n = 1:length(ns)
    FN(n) = (1/T) * trapz(t, y .* exp(-1j * (2*pi/T) * ns(n) * t));
end

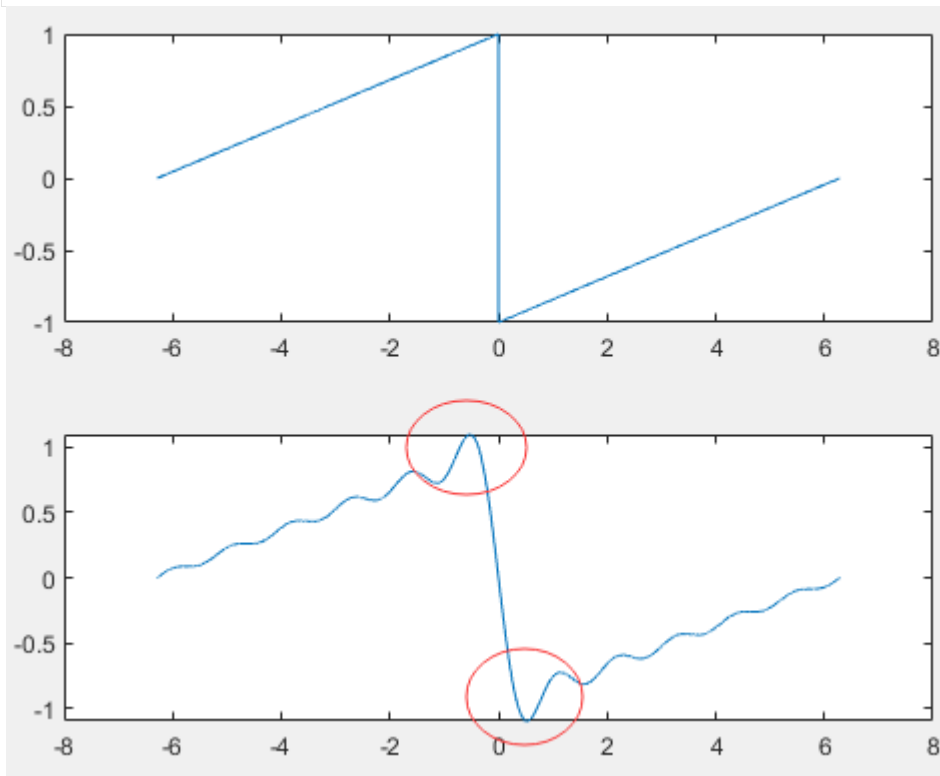
% 重建信号
ft = zeros(size(t));
for n = 1:length(ns)
    ft = ft + FN(n) * exp(1j * (2*pi/T) * ns(n) * t);
end

% 画出原始信号和重建信号

subplot(2,1,1) % 用于同时展示两个图
plot(t,y);
subplot(2,1,2) % 用于同时展示两个图
plot(t,ft) % 还原的 ft 应该是实的, 只画实部

```

在周期信号的情况下, 吉布斯现象通常会在信号的间断处出现, 已用红框框出



-11 到 11 的傅里叶系数如下：

列 1 至 9

0.0001 - 0.0289i 0.0001 - 0.0318i 0.0001 - 0.0354i
 0.0001 - 0.0398i 0.0001 - 0.0455i 0.0001 - 0.0531i 0.0001
 - 0.0637i 0.0001 - 0.0796i 0.0001 - 0.1061i

列 10 至 18

0.0001 - 0.1592i 0.0001 - 0.3183i 0.0001 + 0.0000i
 0.0001 + 0.3183i 0.0001 + 0.1592i 0.0001 + 0.1061i 0.0001
 + 0.0796i 0.0001 + 0.0637i 0.0001 + 0.0531i

列 19 至 23

0.0001 + 0.0455i 0.0001 + 0.0398i 0.0001 + 0.0354i
 0.0001 + 0.0318i 0.0001 + 0.0289i

实验总结（完成心得与其它，主要讲自己在实验中遇到的问题和解决方法）：
 本次实验学会了 **MATLAB** 的积分，也了解了 **fft** ()，**ifft** () 等函数。

在实验的最开始，本来希望使用 **fft** 函数完成,再用 **ifft** 还原。

但是查阅资料显示，**fft** 函数返回的是信号的频域表示，而不是算出傅里叶级数。并且 **fft** 返回的数据量很大，对我这种新手来说，从返回的结果里面提取前 11 次谐波的内容总是出错。

并且实验要求编程计算出傅里叶级数。从频域表示 转换到 傅里叶级数，也比较困难，所以最终选择了自己积分，一个个算出 **Fn**，再把谐波累加，还原回去。

参考文献（包括参考的书籍，论文，URL 等，重要）：

在 MATLAB 中，`sawtooth` 函数用于生成锯齿波形。锯齿波形是一种周期性信号，其波形呈锯齿状，类似于锯齿。这种波形通常由周期性的上升和下降的直线段组成，类似于音乐中的吉他演奏时的音调变化。

以下是 MATLAB 中 `sawtooth` 函数的基本语法：

```
y = sawtooth(t)
```

其中，`t` 是时间向量，`y` 是与时间向量 `t` 对应的锯齿波形。默认情况下，`sawtooth` 函数生成的波形周期为 2π ，幅度为 1，相位为 0。但是，你可以通过提供额外的参数来控制周期、幅度和相位。

例如可以使用以下形式的函数调用来生成特定参数的锯齿波形：

```
y = sawtooth(t, amplitude)
```

```
y = sawtooth(t, frequency)
```

```
y = sawtooth(t, phase)
```

其中，`amplitude` 是波形的幅度，`frequency` 是波形的频率（周期的倒数），`phase` 是波形的相位偏移。

在 MATLAB 中，`fft` 函数计算的是离散傅立叶变换（DFT），其结果是采用指数形式的。具体来说，DFT 的结果是一组复数，其中包含了信号在离散频率点上的幅度和相位信息。

对于离散信号 $x(n)$ ，其 DFT $X(k)$ 计算公式如下所示：

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}$$

其中， N 是信号的长度， $x(n)$ 是输入信号， $X(k)$ 是 DFT 结果， k 是频率索引 ($0 \leq k < N$)， j 是虚数单位。

这个公式中的指数项 $e^{-j2\pi nk/N}$ 表示了每个频率分量的相位信息。FFT 函数计算的结果 Y 也是一个复数向量，其中包含了信号在离散频率点上的幅度和相位信息。

实验题目：

2. Write a function called **square_wave** that computes the sum

$$\sum_{k=1}^n \frac{\sin[(2k-1)t]}{2k-1}$$

for each of 1001 values of t uniformly spaced from 0 to 4π inclusive. The input argument is a scalar non-negative integer n , and the output argument is a row vector of 1001 such sums—one sum for each value of t . You can test your function by calling it with $n = 20$ or greater and plotting the result and you will see why the function is called “square_wave”.

完成英文部分的实验, 读出 $n = 200$ 时 square_wave 函数生成的波形的参数(如幅度、周期等), 利用该参数和 MATLAB 函数 square()画出一致的标准波形。

实验摘要 (简单描述实验内容):

在不同的 t , 不同的 n 的情况下, 计算这个级数的值, 并画图

题目描述 (以中文简要描述实验题目, 建议陈述自己对题目的理解):

T 是一个向量, 从 $0-4\pi$, 步长是 $4\pi/1000$, 其长度应该是 1001

n 是一个非负的整数标量

输出表达式是一个长度为 1001 的行向量, 记为 rst

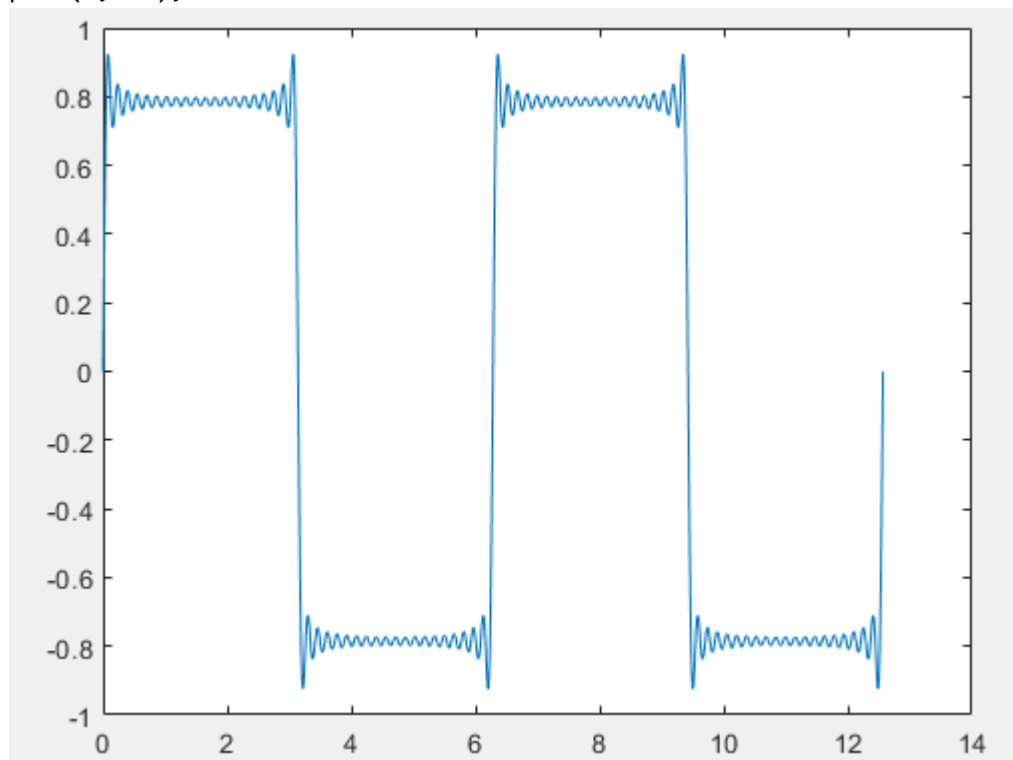
对 T 的每一个分量 t , n 取 n 时, rst 的对应分量就是上面这个级数值。

然后, 题目让我们 n 取 20 或者更大的情况下, 跑一下程序。并且画出 rst , 看看这个函数为什么叫“方波”

实验过程与分析 (详细实验内容):

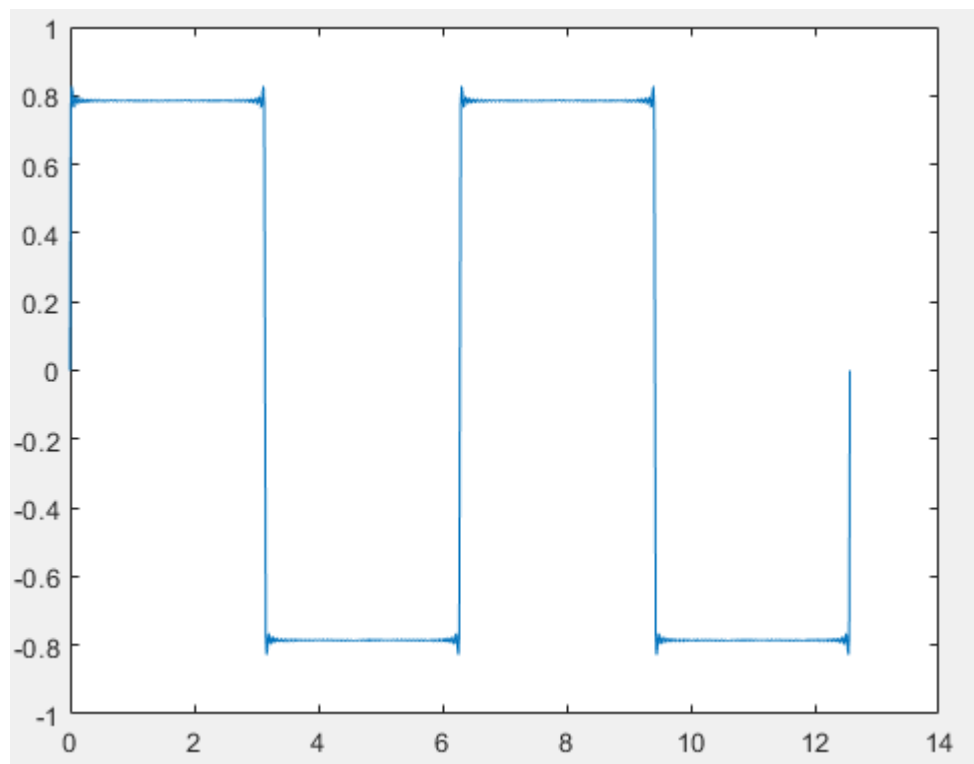
MATLAB 代码:

```
t = 0:(4*pi)/1000:4*pi;  
  
N = 20;  
  
rst = zeros(1,length(t));%创建一个与 t 长度相等的行向量  
for n = 1:N  
    rst = rst + (sin((2*n-1).*t))/(2*n-1);  
end  
  
plot(t,rst);
```



上图为 $n=20$ 时，可以看到，这类似一个方波信号。

现在令 $n=200$ ，读出 $n = 200$ 时 `square_wave` 函数生成的波形的参数(如幅度、周期等)，



观察到幅值大概是 **0.8**，周期是 2π 。占空比是 **50%**

现在利用 `square` 画出标准方波

```
t = 0:(4*pi)/1000:4*pi;
```

```
N = 200;
```

```
rst = zeros(1,length(t));%创建一个与 t 长度相等的行向量
```

```
for n = 1:N
```

```
    rst = rst + (sin((2*n-1).*t))/(2*n-1);
```

```
end
```

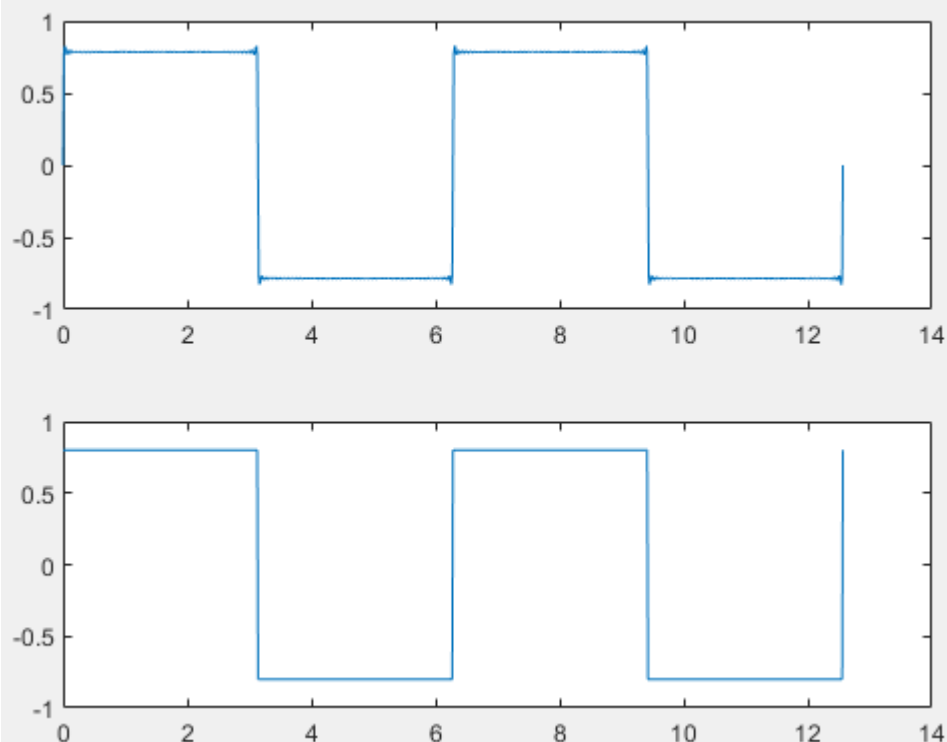
```
g = 0.8 .* square(t, 50);%方波默认周期正好是  $2\pi$ ，默认振幅是 1，故乘上 0.8
```

```
subplot(2,1,1)%用于同时展示两个图
```

```
plot(t,rst);
```

```
subplot(2,1,2)%用于同时展示两个图
```

```
plot(t,g);
```



可以看到 $n=200$ 时，谐波叠加出的方波与标准方波已经非常相似

实验总结（完成心得与其它，主要讲自己在实验中遇到的问题 and 解决方法）：

本次实验，体会到了傅里叶级数的意义，观察到了吉布斯现象（在方波的拐角处），也锻炼了 **matlab** 的编程能力，顺便还学了下英语

这一题，实验很顺利，没有遇到什么不会的地方和 **bug**。

参考文献（包括参考的书籍，论文，URL 等，重要）：

在MATLAB中，`square` 函数用于生成一个方波信号。该函数的基本语法是：

matlab

Copy code

```
y = square(t)
```

其中，`t` 是时间向量，`y` 是生成的方波信号。

默认情况下，`square` 函数生成的方波信号周期为 2π ，振幅为 1，duty cycle（占空比）为 50%。可以通过提供额外参数来调整这些属性。例如：

matlab

Copy code

```
y = square(t, duty)
```

其中，`duty` 是占空比，范围在 0 到 100 之间，默认为 50。

方波信号的周期由 `t` 的范围决定，因此，如果 `t` 范围是 0 到 4π ，那么生成的方波信号将具有 2π 的周期。

`square` 函数有一些可选参数，可以用来调整生成的方波信号的属性。主要的可选参数包括：

- Duty Cycle（占空比）**：这个参数控制方波信号的脉冲宽度与周期之比。占空比为 50% 时，方波信号的脉冲宽度等于周期的一半。可以通过在函数中提供第二个参数来设置占空比，例如 `y = square(t, duty)`，其中 `duty` 是占空比，范围在 0 到 100 之间，默认为 50。
- Phase Shift（相位移）**：这个参数控制方波信号的相位移。可以通过在函数中提供第三个参数来设置相位移，例如 `y = square(t, duty, phase)`，其中 `phase` 是相位移，以弧度为单位，默认为 0。
- Amplitude（振幅）**：方波信号的默认振幅是 1。可以通过在函数中提供第四个参数来设置振幅，例如 `y = square(t, duty, phase, amplitude)`，其中 `amplitude` 是振幅，默认为 1。

实验题目:

用 **MATLAB** 进行以下实验，回答问题并粘贴实验过程中产生的结果图。

1. What function $f(t)$ has the Fourier series

$$\sum_{n=1}^{\infty} \frac{\sin nt}{n}$$

You can evaluate the sum analytically or numerically. Either way, guess a closed form for $f(t)$ and then sketch it.

2. Confirm your conjecture for $f(t)$ by finding the Fourier series coefficients f_n for $f(t)$. Compare your result to the expression in the previous part. What happens to the cosine terms?
3. Define the partial sum

$$f_N(t) = \sum_{n=1}^N \frac{\sin nt}{n}$$

Plot some $f_N(t)$'s. By what fraction does $f_N(t)$ overshoot $f(t)$ at worst? Does that fraction tend to zero or to a finite value as $N \rightarrow \infty$? If it is a finite value, estimate it. (hint: Gibbs phenomenon)

4. Now define the average of the partial sums:

$$F_N(t) = \frac{f_1(t) + f_2(t) + f_3(t) + \cdots + f_N(t)}{N}$$

Plot some $F_N(t)$'s. Compare your plots with those of $f_N(t)$ that you made in the previous part, and **qualitatively** explain any differences.

实验摘要

题目描述 (以中文简要描述实验题目, 建议陈述自己对题目的理解):

实验过程与分析（详细实验内容）：

一 实验基本原理及步骤（理论计算，原理：必要的公式，图表；步骤，如有必要画出流程图，应给出主要实现步骤代码）

二 实验结果

三 实验结果分析

实验总结（完成心得与其它，主要讲自己在实验中遇到的问题 and 解决方法）：

参考文献（包括参考的书籍，论文，URL 等，重要）: