

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 1 日

**实验题目：**

1. 利用MATLAB实现下列信号，并绘出图形

(1)  $f(t) = \varepsilon(t)$ ，取  $t = 0 \sim 10$

(2)  $f(t) = 4e^{-0.5t} \cos(\pi t)$ ，取  $t = 0 \sim 10$

(3)  $f(k) = \varepsilon(k+2) - \varepsilon(k-5)$

(4)  $f(k) = 7(0.6)^k \cos(0.9\pi k)$

**实验摘要**

这一题，要求使用 **MATLAB** 画出信号的图形。

**题目描述**

一二两问要求绘制出连续信号， $t$  取 **0-10**.但是计算机中是用离散来近似连续， $t$  的步长可能会对图形产生较大的影响。

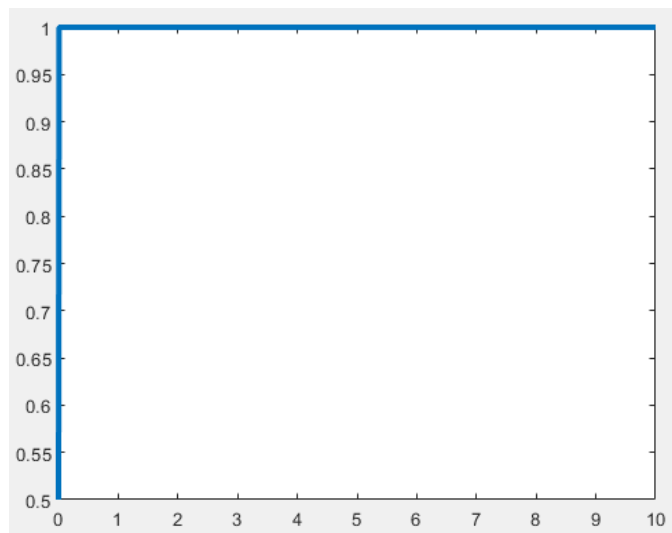
三四两问，要求绘制离散型信号，直接画就可以了。

## 实验过程与分析

### (1)

先试试  $\Delta t$  的步长是 **0.01** 的情况下，绘制的图形

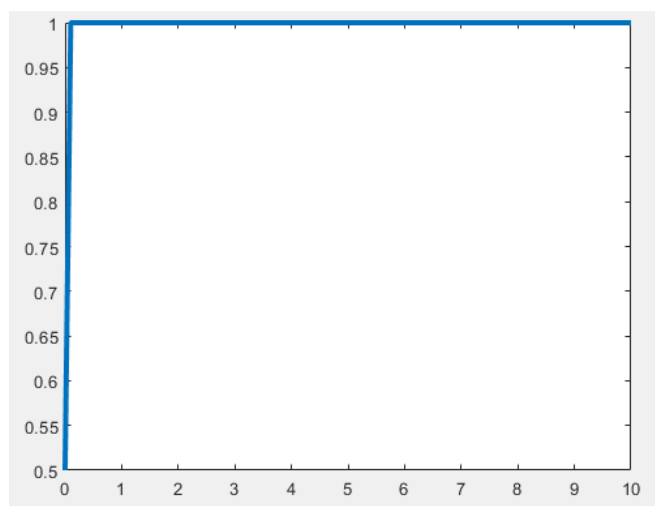
```
t = 0:0.01:10;  
u = heaviside(t);  
plot(t,u);
```



可以看到，几乎是垂直着上去的。

再试试  $\Delta t$  的步长是 **0.1** 的情况下

```
t = 0:0.1:10;  
u = heaviside(t);  
plot(t,u);
```



可以看到，步长 **0.1** 下，并不是理想的阶跃信号，它是有一定斜率的

(2)

## MATLAB 代码:

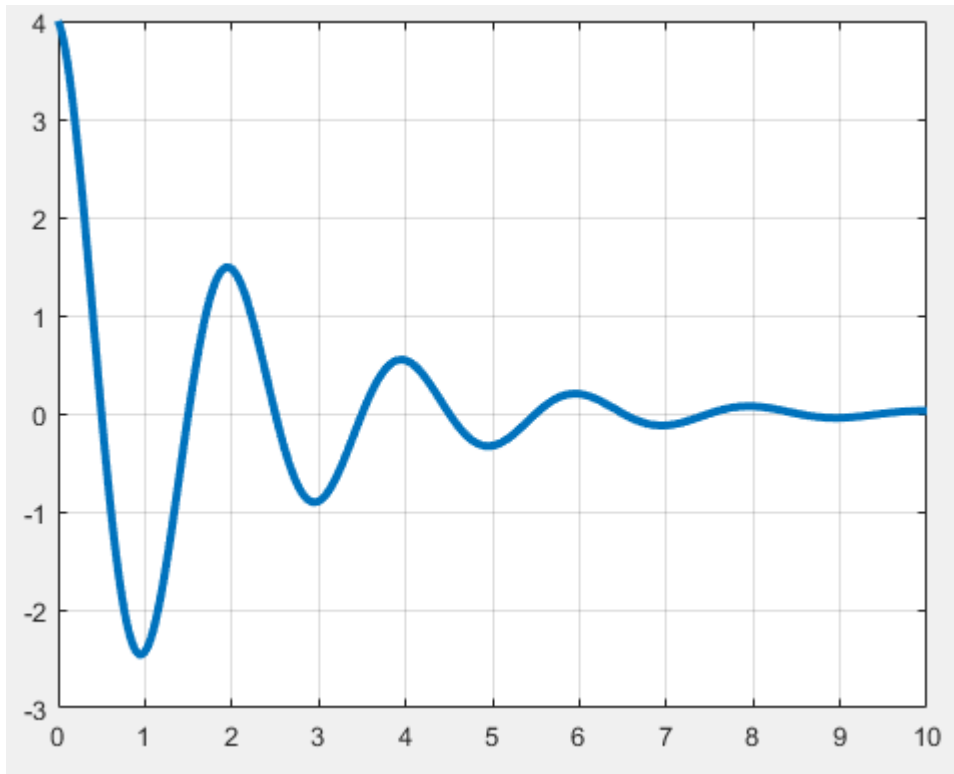
```
t = 0:0.01:10;
```

西安电子科技大学网络与信息安全学院实验报告

```
u = 4 * exp(-0.5*t) .* cos(pi*t);
```

```
plot(t,u,'LineWidth',3);
```

```
grid on;%用于显示方格线
```



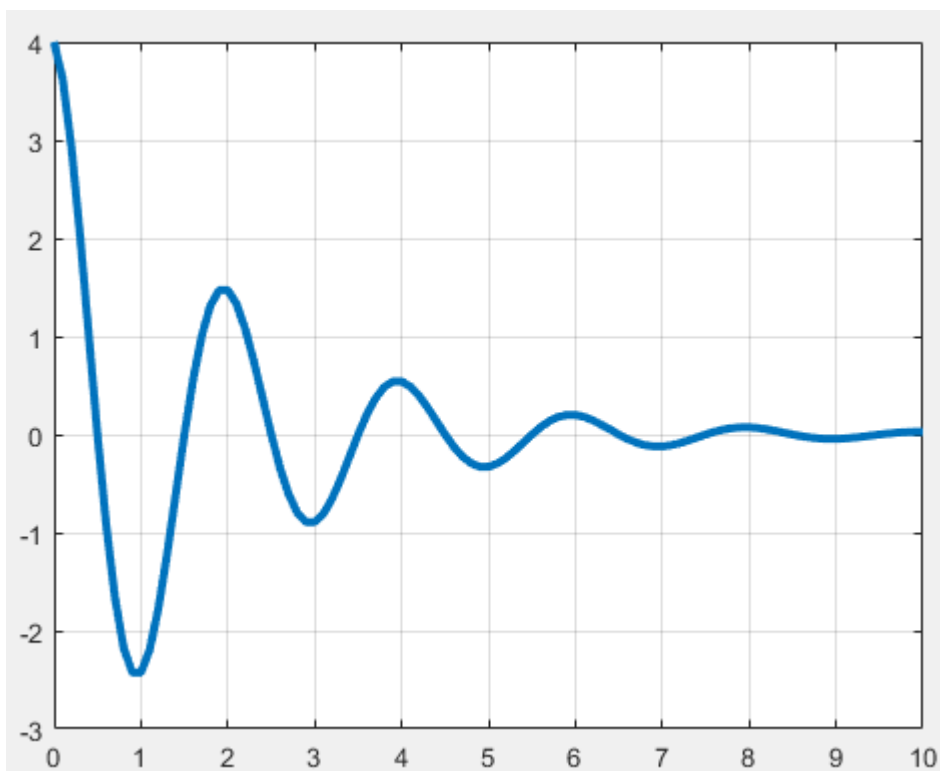
调整一下步长，变为 **0.1**，发现几乎没有区别

```
t = 0:0.1:10;
```

```
u = 4 * exp(-0.5*t) .* cos(pi*t);
```

```
plot(t,u,'LineWidth',3);
```

```
grid on;%用于显示方格线
```



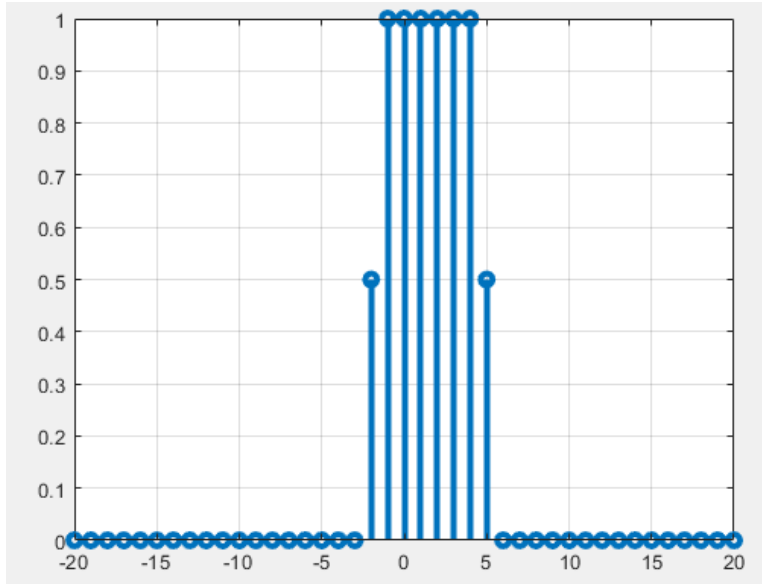
(3)

**MATLAB 代码:**

```

k = -20:20
f1 = heaviside(k+2)
f2 = heaviside(k-5)
stem(k,f1-f2,'LineWidth',3);
grid on;

```



(4)

**MATLAB 代码:**如果  $k$  是从 -20 到 20

```

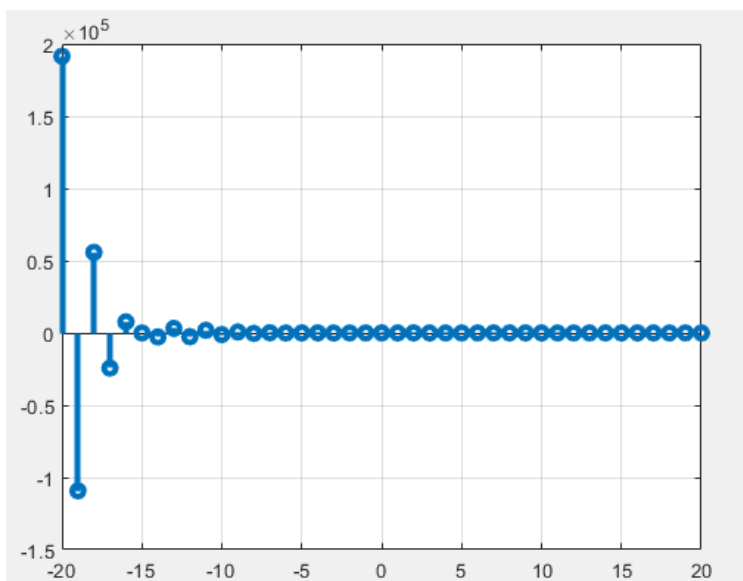
k = 0:20
f = 7 .* 0.6.^k .* cos(0.9*pi*k)

```

```

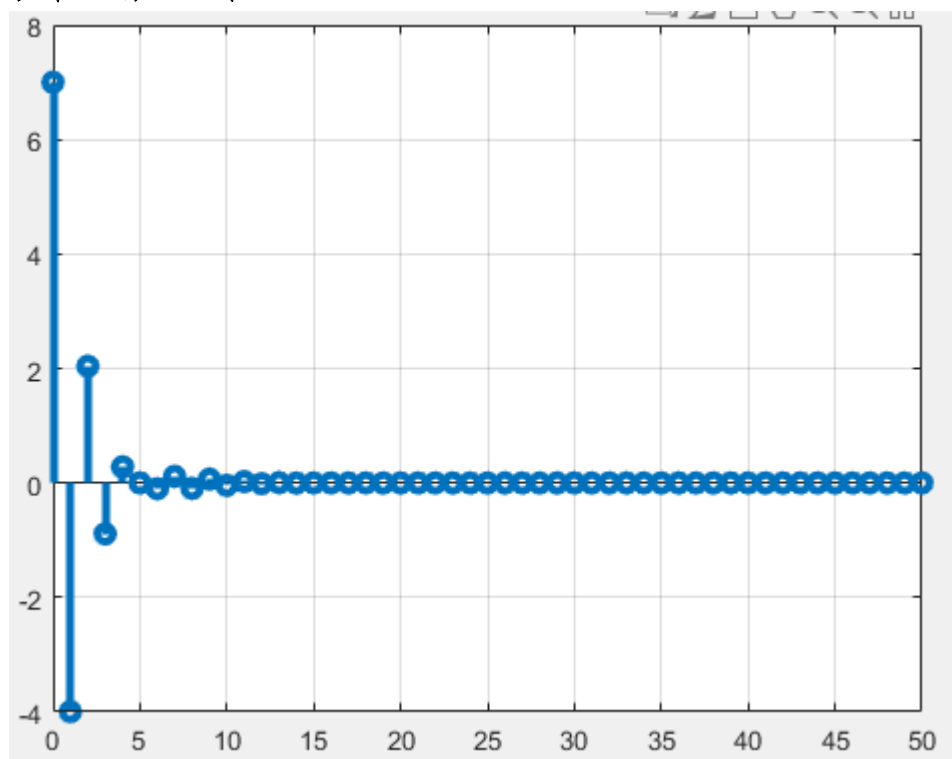
stem(k,f,'LineWidth',3);
grid on;

```



可以看到，**-20** 时候已经达到了 **200000**，后面的函数值相对于 **200000** 非常小，几乎贴在 **t** 轴上，所以换一个 **k** 的区间再试试

现在 **k** 从 **0** 到 **50**



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）：

**MATLAB** 绘制连续性图像时，是用离散数据来拟合的，因此离散数据的步长对图像影响很大，在实验中切换步长，可以看到不同的结果

离散型信号的 **k** 取值对图像也有影响，如果 **k** 取的不合适，生成的图像没准会很差

通过这个题目，我初步掌握了利用 **matlab** 绘制信号（函数）图像的方法，体会到了到了实际阶跃信号在 **0** 处并不是理想的。

参考文献（包括参考的书籍，论文，URL 等，重要）：

**MATLAB 绘制图像的参考文献**

[https://blog.csdn.net/qq\\_44486550/article/details/105276774?ops\\_request\\_misc=%257B%2522request%255Fid%2522%253A%2522171517703316800182138422%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\\_id=171517703316800182138422&biz\\_id=0&utm\\_medium=distribute.pc\\_search\\_result.none-task-blog-2~all~sobaiduend~default-2-105276774-null-null.142^v100^pc\\_search\\_result\\_base4&utm\\_term=matlab%E7%BB%98%E5%88%B6%E7%A6%BB%E6%95%A3%E5%9B%BE%E5%83%8F&spm=1018.2226.3001.4187](https://blog.csdn.net/qq_44486550/article/details/105276774?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171517703316800182138422%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=171517703316800182138422&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-2-105276774-null-null.142^v100^pc_search_result_base4&utm_term=matlab%E7%BB%98%E5%88%B6%E7%A6%BB%E6%95%A3%E5%9B%BE%E5%83%8F&spm=1018.2226.3001.4187)

**MATLAB 产生信号的参考文献**

[https://blog.csdn.net/weixin\\_69250798/article/details/125297089?ops\\_request\\_misc=%257B%2522request%255Fid%2522%253A%2522171517379516800185878385%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\\_id=171517379516800185878385&biz\\_id=0&utm\\_medium=distribute.pc\\_search\\_result.none-task-blog-2~all~baidu\\_landing\\_v2~default-2-125297089-null-null.142^v100^pc\\_search\\_result\\_base4&utm\\_term=matlab%E8%A1%A8%E7%A4%BA%E9%98%B6%E8%B7%83%E4%BF%A1%E5%8F%B7&spm=1018.2226.3001.4187](https://blog.csdn.net/weixin_69250798/article/details/125297089?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522171517379516800185878385%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=171517379516800185878385&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~baidu_landing_v2~default-2-125297089-null-null.142^v100^pc_search_result_base4&utm_term=matlab%E8%A1%A8%E7%A4%BA%E9%98%B6%E8%B7%83%E4%BF%A1%E5%8F%B7&spm=1018.2226.3001.4187)

实验题目：

2. 某系统满足的微分方程为

$$y''(t) + 4y'(t) + 3y(t) = 2f'(t) + f(t)$$

(1) 利用MATLAB求系统的单位冲激响应，并绘出图形

(2) 利用MATLAB求系统的单位阶跃响应，并绘出图形

参考函数：tf(), impulse(), step()

实验摘要（简单描述实验内容）：

本题要求用 **matlab** 求出一个系统的两个基本响应，并绘制出基本响应的图形

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）：

首先用 **tf()**函数创建一个 “传递函数对象”

这个传递函数似乎要用拉普拉斯变换来得到，我通过查阅资料，了解到了这个微分方程的传递函数是什么

然后用 **Impulse().step()**分别求其冲激、阶跃响应

最后画图，就完事了

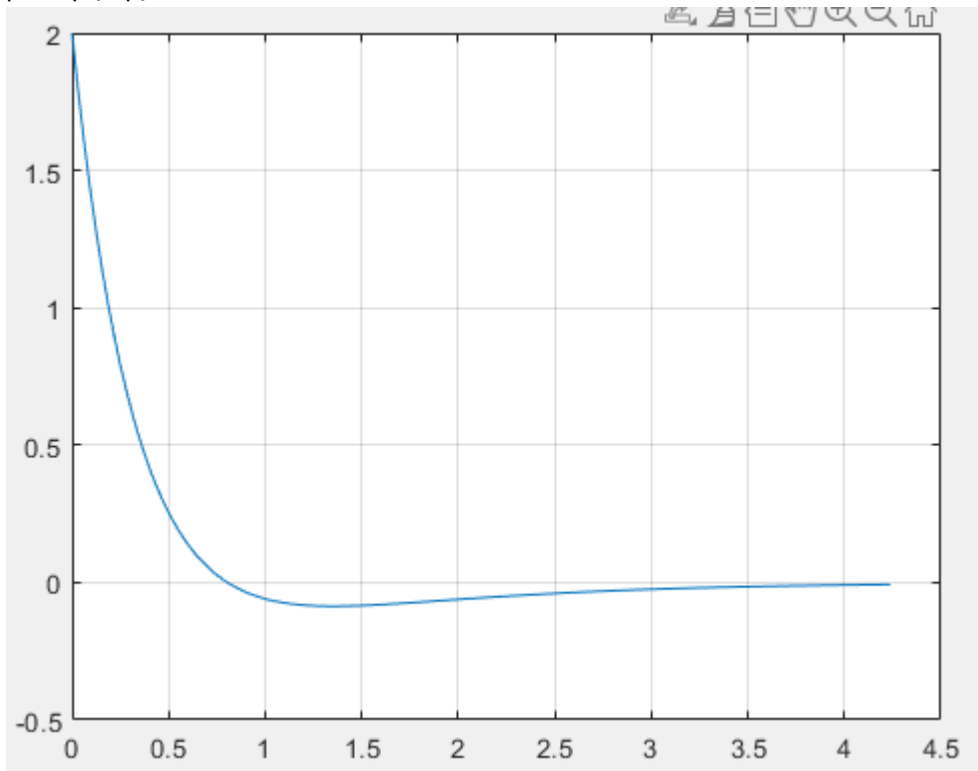


## 实验过程与分析

### (1)

查阅资料，已经知道了传递函数的分子分母，按部就班即可

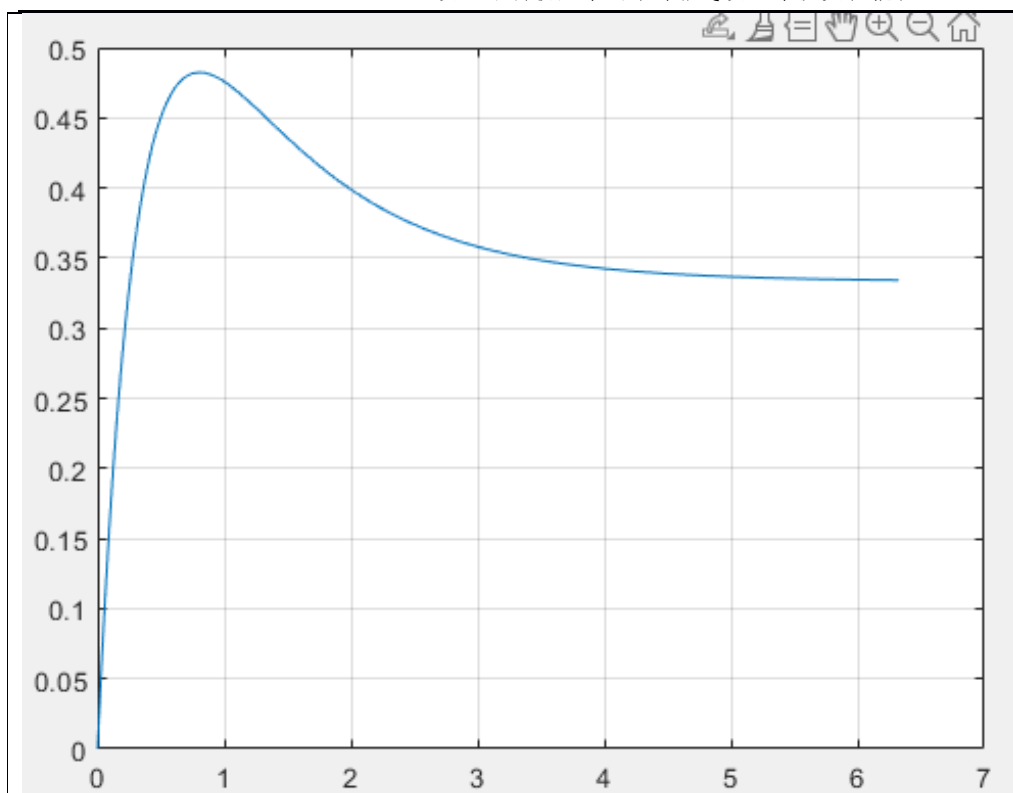
```
num = [2, 1];  
den = [1, 4, 3];  
H = tf(num, den);  
[h, t] = impulse(H);  
plot(t,h);
```



### (2)

把上一问的 **impulse** 改成 **step** 就行了

```
num = [0, 2, 1];  
den = [1, 4, 3];  
H = tf(num, den);  
[h, t] = step(H);  
plot(t,h);  
grid on;
```



### 实验总结

我先查询 **tf** 函数的说明，里面有一个“传递函数对象”不知道是什么东西  
然后去查询“传递函数对象”，了解到这东西与拉普拉斯变换有关

然后根据资料，算出了传递函数对象的分子分母

最后用手解一下微分方程，**MATLAB** 计算的结果是正确的。

## 参考文献（包括参考的书籍，论文，URL 等，重要）：

### Tf 的说明

在 MATLAB 中，`tf` 函数用于创建传递函数对象。传递函数是用于描述线性时不变系统的数学模型。

`tf` 函数的基本语法如下：

```
matlab
```

[Copy code](#)

```
H = tf(num, den)
```

其中，`num` 和 `den` 分别是传递函数的分子和分母多项式的系数向量。这两个参数可以是向量或矩阵，分别代表单输入单输出（SISO）系统和多输入多输出（MIMO）系统。

下面是一个简单的示例，创建一个传递函数对象：

```
matlab
```

[Copy code](#)

```
num = [1];  
den = [1 2 1];  
H = tf(num, den);
```

这个例子创建了一个传递函数对象 `H`，其传递函数为  $\frac{1}{s^2+2s+1}$ 。

### Impulse 和 step 的说明

在 MATLAB 中，`impz` 和 `step` 是用于分析线性时不变系统的两个常用函数，用于计算系统对单位冲激和单位阶跃输入的响应。

#### 1. impulse 函数：

`impz` 函数用于计算系统对单位冲激输入的响应。其基本语法如下：

```
matlab
```

[Copy code](#)

```
[y, t] = impulse(sys)
```

其中，`sys` 是传递函数、状态空间模型或频率响应对象。函数返回的 `y` 是系统对单位冲激输入的响应，而 `t` 是相应的时间向量。例如，对于一个传递函数对象 `H`，可以这样计算其对单位冲激输入的响应：

```
matlab
```

[Copy code](#)

```
H = tf([1], [1, 2, 1]);  
[y, t] = impulse(H);  
plot(t, y);
```

#### 2. step 函数：

`step` 函数用于计算系统对单位阶跃输入的响应。其基本语法如下：

```
matlab
```

[Copy code](#)

```
[y, t] = step(sys)
```

同样，`sys` 是传递函数、状态空间模型或频率响应对象。函数返回的 `y` 是系统对单位阶跃输入的响应，而 `t` 是相应的时间向量。例如，对于一个传递函数对象 `H`，可以这样计算其对单位阶跃输入的响应：

```
matlab
```

[Copy code](#)

```
H = tf([1], [1, 2, 1]);  
[y, t] = step(H);  
plot(t, y);
```

## 传递函数的说明

### 1. 传递函数:

传递函数是用于描述线性时不变系统的数学模型。对于一个系统，其传递函数  $H(s)$  是输入  $f(t)$  和输出  $y(t)$  的关系，其中  $s$  是复变量。传递函数可以表示为输出和输入的拉普拉斯变换的比值。传递函数通常在复平面上的某些点上具有物理意义，例如，它们的极点和零点可以提供关于系统的稳定性和动态响应的信息。

### 2. 拉普拉斯变换:

拉普拉斯变换是一种数学工具，用于将时域中的函数转换为复频率域中的函数。对于一个信号  $f(t)$ ，其拉普拉斯变换  $F(s)$  定义为积分：

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^{\infty} e^{-st} f(t) dt$$

其中， $s$  是复变量。拉普拉斯变换将一个时域函数转换为一个复平面上的复函数，其中复平面的实部和虚部分别对应于信号的幅度和相位。通过拉普拉斯变换，可以将微分和积分等运算转换为简单的代数运算，使得对线性时不变系统的分析更加方便。

### 3. 传递函数和拉普拉斯变换的关系:

传递函数和拉普拉斯变换之间的关系在于，传递函数是系统的拉普拉斯变换的比值。对于一个线性时不变系统，其传递函数  $H(s)$  可以表示为输出  $Y(s)$  和输入  $F(s)$  的比值：

$$H(s) = \frac{Y(s)}{F(s)}$$

其中， $Y(s)$  和  $F(s)$  分别是系统的输出和输入的拉普拉斯变换。传递函数描述了系统对不同频率的输入信号的响应特性，例如，它可以告诉我们系统对特定频率的信号是放大还是衰减，以及是否存在特定的频率响应。

实验题目：

3. 利用MATLAB求下列函数的卷积，并绘制出图形

$$(1) f_1(t) = \varepsilon(t) - \varepsilon(t-1), \quad f_2(t) = 2t[\varepsilon(t) - \varepsilon(t-1)]$$

$$(2) f_1(t) = \cos(30t)g_5(t), \quad f_2(t) = \varepsilon(t) - \varepsilon(t-4)$$

参考函数：conv()

实验摘要

求卷积，并绘制图形

题目描述：

阶跃函数的表达，在第一题中已经练习了。

现在只需查阅 **conv()** 函数的资料，求卷积，绘制图形

## 实验过程与分析

## 源代码：

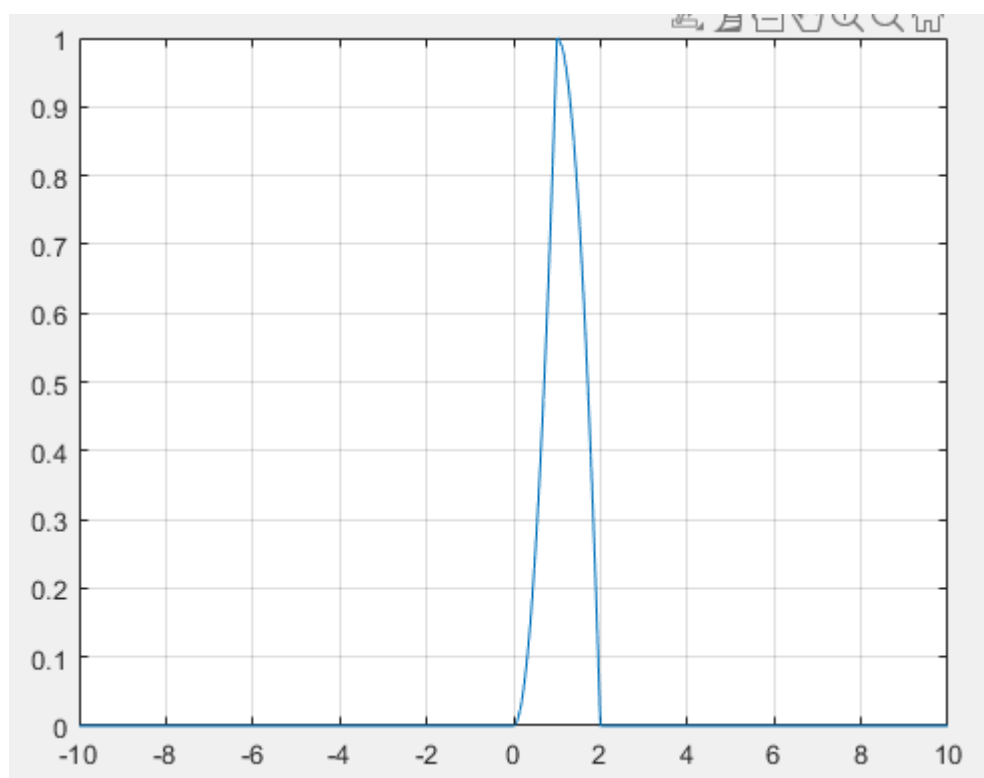
```

t = -10:0.01:10;

f1 = heaviside(t) - heaviside(t-1);
f2 = 2 .* t .* f1;
result = conv(f1,f2,'same');
result = result .* 0.001;
plot(t,result);
grid on;

```

我们必须在 **conv** 函数中使用 ‘same’，不然 **result** 和 **t** 的长度不一致，没法画图



可以观察到图形出现了一些变化，而且纵轴也发生了变化

(2)

$$(2) f_1(t) = \cos(30t)g_5(t), \quad f_2(t) = \varepsilon(t) - \varepsilon(t-4)$$

不知道这个 **g5** 是什么，所以无从下手

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）：

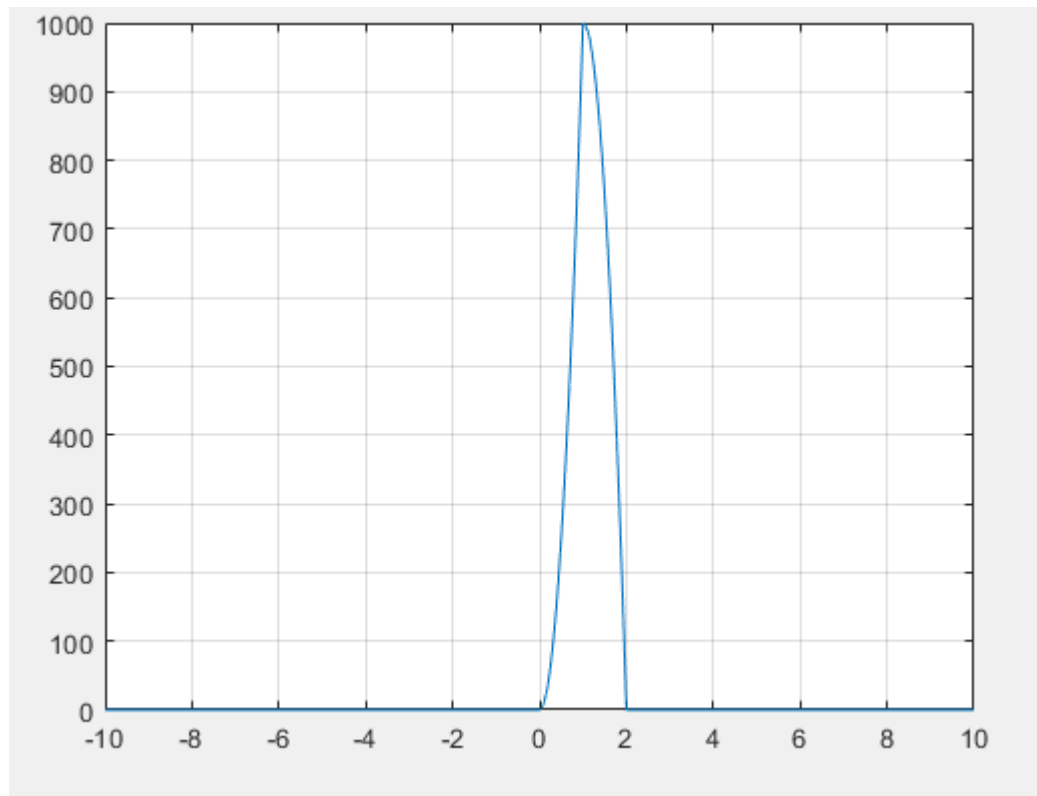
用纸笔计算，发现卷积结果应该是

$$t^2\varepsilon(t) - 2(t-1)^2\varepsilon(t-1) + (t-2)^2\varepsilon(t-2) - 2(t-1)\varepsilon(t-1) + 2(t-2)\varepsilon(t-2)$$

在各个区间，应该是二次函数型，

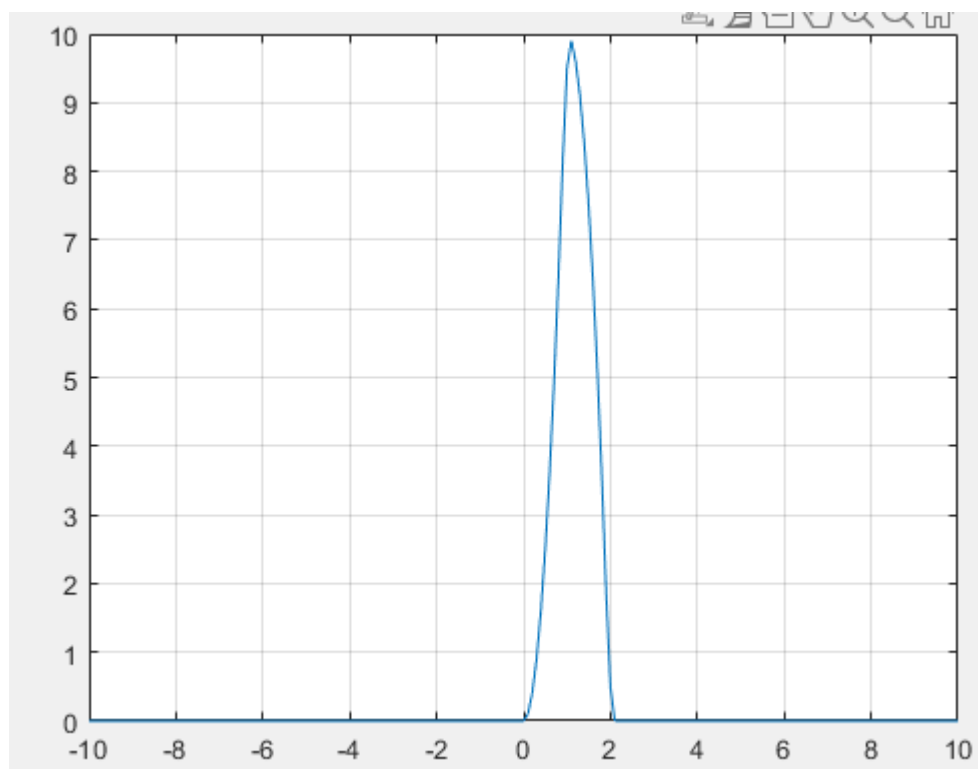
但是在一开始实验中，步长不同会导致纵坐标也不一样，最初推测是阶跃函数的突变导致的。如下：

步长改为 **0.001**



图像的顶点又变到了 **1000**

步长为 **0.1**：



后来查阅资料发现

在 MATLAB 中，卷积操作会对相乘后的结果进行积分。但在这种情况下，我们实际上需要对积分结果再乘以步长才能得到正确的结果。原因在于，卷积的离散形式可以看作是对两个函数的乘积进行积分，但在离散情况下，积分就变成了求和，因此需要将结果乘以步长来近似替代积分。

于是修改代码,给结果乘以步长，最终得到了正确答案。

参考文献（包括参考的书籍，论文，URL 等，重要）：

## Conv 函数的文献

`conv()` 函数是 MATLAB 中用于计算离散和连续卷积的函数之一。卷积是一种在信号处理和系统分析中广泛使用的数学运算，用于描述两个信号之间的相互影响。在 MATLAB 中，`conv()` 函数可以用于计算离散信号和连续信号的卷积。

### 1. 离散卷积：

对于离散信号，`conv()` 函数的基本语法是：

matlab

Copy code

```
c = conv(a, b)
```

其中 `a` 和 `b` 是要进行卷积的两个向量，`c` 是计算得到的卷积结果。例如：

matlab

Copy code

```
a = [1, 2, 3];  
b = [4, 5];  
c = conv(a, b);
```

这将计算向量 `a` 和 `b` 的卷积，并将结果存储在向量 `c` 中。

### 2. 连续卷积：

对于连续信号，`conv()` 函数的语法类似，但是参数和输出的含义略有不同。其基本语法是：

matlab

Copy code

```
c = conv(u, v, 'full')
```

其中 `u` 和 `v` 是要进行卷积的两个函数，`c` 是计算得到的卷积结果。`'full'` 参数指示 MATLAB 输出完整的卷积结果。例如：

matlab

Copy code

```
u = @(t) exp(-t);  
v = @(t) sin(t);  
t = linspace(0, 10, 100);  
c = conv(u(t), v(t), 'full');
```

这将计算函数 `u(t)` 和 `v(t)` 的卷积，并将结果存储在向量 `c` 中。



实验题目:

4. 利用 MATLAB 产生高斯白噪声, 绘出图形, 并求其自相关函数, 绘出图形。

参考函数: `randn()`, `wgn()`, `xcorr()`, `autocorr()`

实验摘要 (简单描述实验内容):

产生高斯白噪声, 和其自相关函数, 并画出图形

题目描述:

产生白噪声有两种方法:

1. **Randn()** 函数可以产生指定长度的随机数向量, 这本身就是高斯白噪声的一种表达形式, 可以直接以 **randn** 的返回结果作为信号。

2. 也可以使用 **wgn()** 函数产生指定长度和功率的高斯白噪声信号

计算自相关也有两种方法:

1. 相同的信号作为两个输入参数给 **xcorr** 函数时, 它就会计算该信号的自相关。**xcorr** 可以不用指定滞后阶数, 他自己会计算出整个滞后范围(默认是  $-N+1$  到  $N-1$ ), 供我们截取

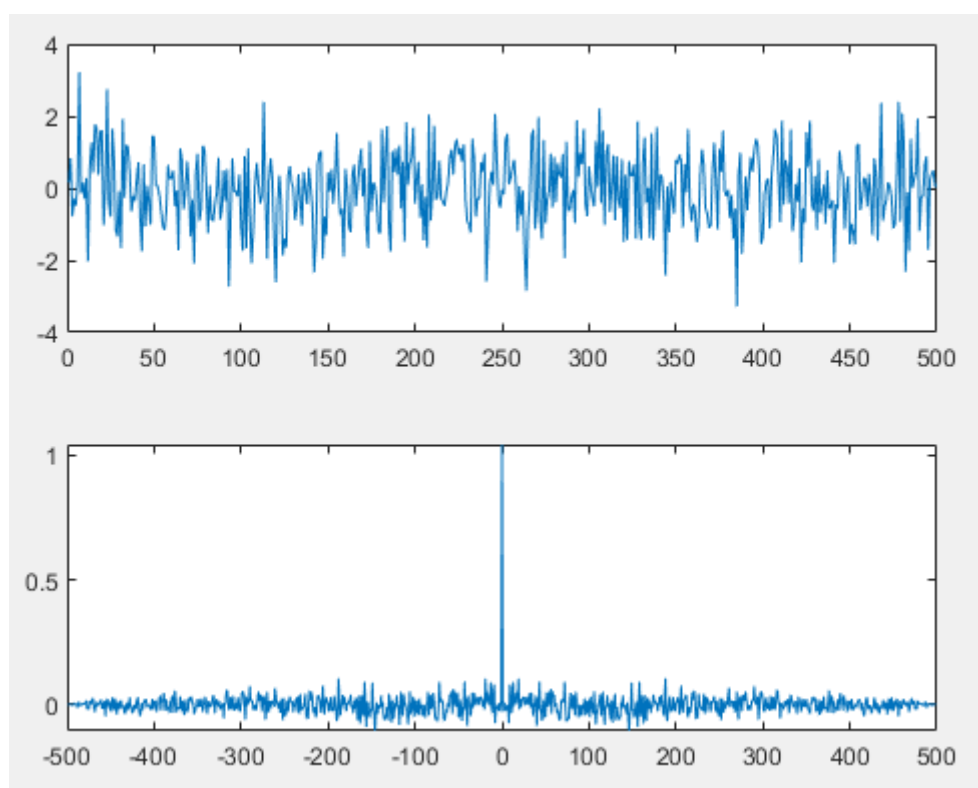
2. **autocorr** 函数直接计算信号的自相关, 但是 **autocorr** 函数的滞后值默认从 0 开始, 无法计算负数滞后, 因此我没有采用这一方法

## 实验过程与分析 (详细实验内容):

### 1. 使用 `randn` 函数产生高斯白噪声, 用 `xcorr` 计算其互相关

```
N = 500;  
signal = randn(1, N);  
lag = -N+1:N-1;%时延范围  
autocorrelation = xcorr(signal,signal,'biased');
```

```
subplot(2,1,1)%用于同时展示两个图  
plot(signal);  
subplot(2,1,2)%用于同时展示两个图  
plot(lag,autocorrelation);
```



峰值为 **1** 左右, 而其他时刻几乎趋近于 **0**

## 2.使用 `wgn` 函数产生高斯白噪声，用 `xcorr` 计算其互相关

`N = 500;`

`signal = wgn(N,1,10);` %产生一个长度为 `N`，功率为 `10` 的列向量来表示信号

`lag = -N+1:N-1;` %时延范围

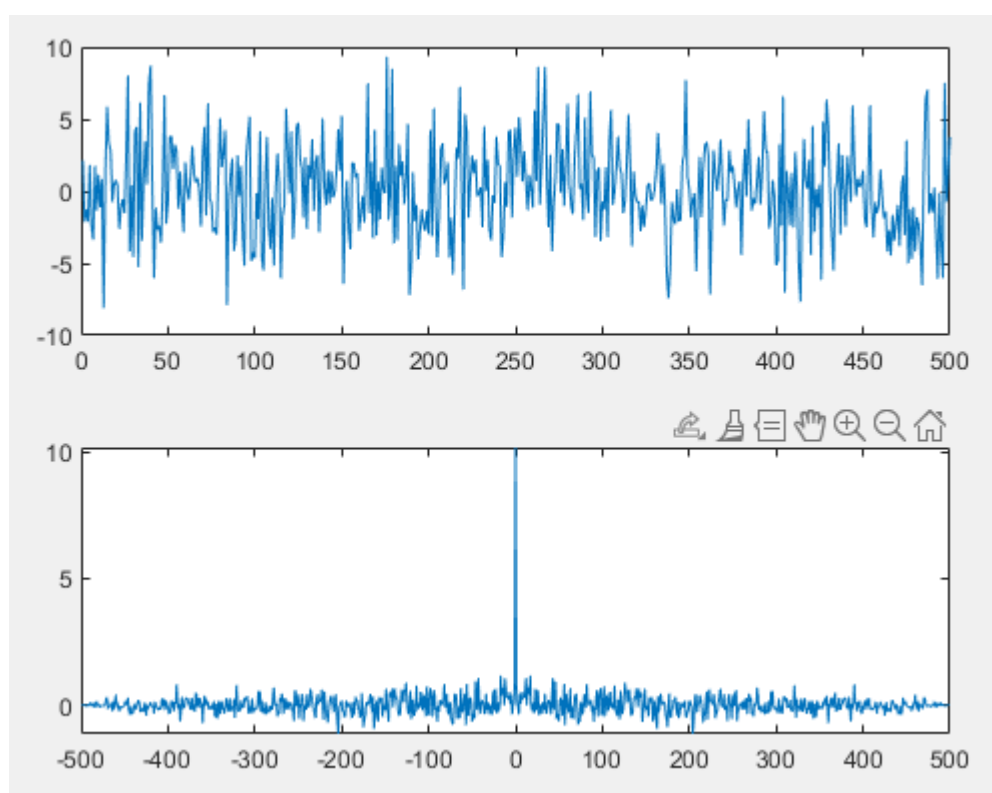
`autocorrelation = xcorr(signal,signal,'biased');`

`subplot(2,1,1)` %用于同时展示两个图

`plot(signal);`

`subplot(2,1,2)` %用于同时展示两个图

`plot(lag,autocorrelation);`



由于产生信号的功率是 **10**，所有自相关函数的峰值也几乎为 **10**，

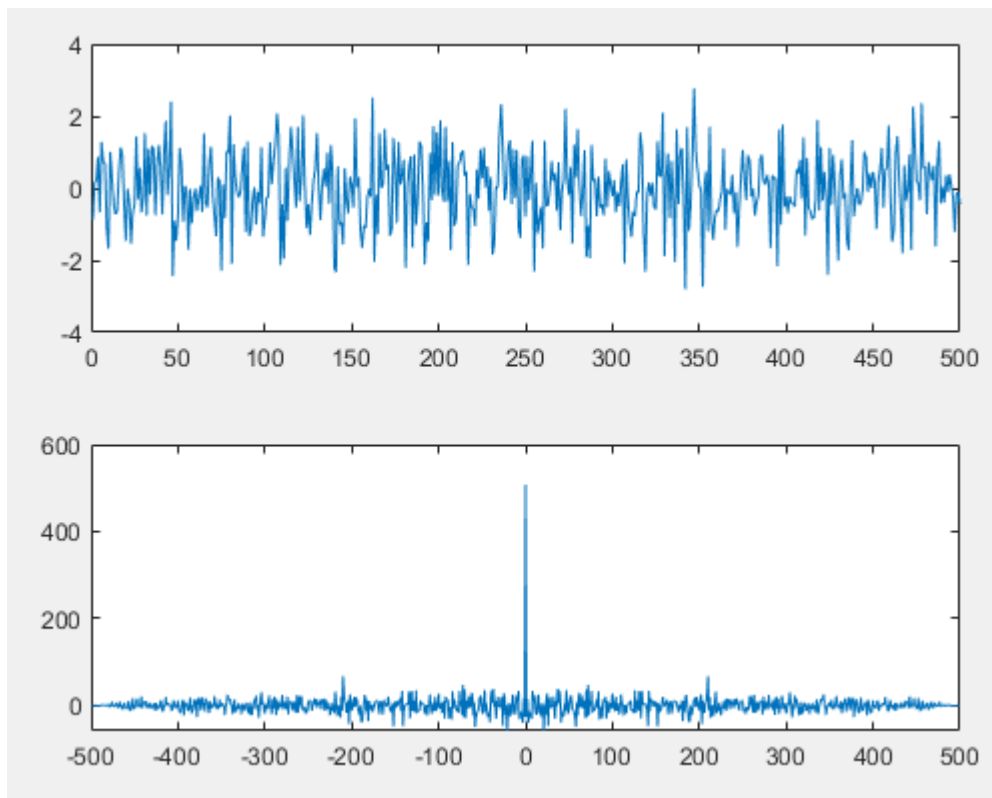
自相关函数符合理论

## 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）：

根据资料：对于高斯白噪声，自相关通常在零延迟处达到峰值，并且在其他延迟值上接近零，

实验中发现，如果 **xcorr** 函数不使用 '**biased**' 参数，则偏差会很大。

一开始不使用此参数，求得的自相关函数为：



峰值接近 **600**，在其他延迟值上也有 **50** 左右的数值，效果不甚理想。

后来查阅资料，使用了 **biased**，实验效果较好。

在 MATLAB 的 '**xcorr**' 函数中，'**biased**' 参数用于指定是否要使用偏差校正来计算自相关或互相关函数。偏差校正通常用于改进样本估计的无偏性。

当 '**biased**' 参数设置为 '**true**' 时，表示要使用偏差校正来计算自相关或互相关函数。这种情况下，自相关或互相关函数的计算公式如下：

$$R[k] = \frac{1}{N} \sum_{n=1}^N x[n]x[n-k]$$

其中  $N$  是时间序列的长度。

当 '**biased**' 参数设置为 '**false**' 时，表示要使用无偏校正来计算自相关或互相关函数。这种情况下，自相关或互相关函数的计算公式如下：

$$R[k] = \frac{1}{N-|k|} \sum_{n=1}^{N-|k|} x[n]x[n-k]$$

通过这一题，我学习到了高斯白噪声、自相关函数，**matlab** 中的这四个函数也练习了 **matlab** 的编程

**参考文献（包括参考的书籍，论文，URL 等，重要）：**

高斯白噪声是一种常见的随机信号，通常用于模拟许多实际系统中的随机干扰或噪声。以下是一些关于高斯白噪声的要点：

1. **高斯性质：** 高斯白噪声是指其在时域内服从高斯分布（也称为正态分布），这意味着其样本值以及样本之间的关系符合正态分布。因此，噪声的任何样本值都是从均值为零、方差为常数的正态分布中独立地随机抽取的。
2. **白噪声性质：** 白噪声的“白”指的是其在频域上的特性，即其功率谱密度在所有频率上均匀分布，因此其频谱呈平坦形状，类似于白光中包含了所有颜色的光谱一样。这意味着白噪声在所有频率上都有相同的能量。
3. **应用：** 高斯白噪声在通信、信号处理、控制系统等领域中有广泛的应用。例如，在通信系统中，它可以模拟由于信道干扰、电子设备本身的噪声等引起的随机干扰；在信号处理中，它可以用作滤波器性能测试、模拟实际环境中的噪声等。
4. **特性：** 高斯白噪声的特性包括平稳性（在统计意义上是平稳的）、独立性（样本之间相互独立）、均值为零等。
5. **生成：** MATLAB 等工具通常提供了生成高斯白噪声的函数。这些函数能够生成服从高斯分布且具有指定均值和方差的随机数序列。

**1. randn:**

- **randn** 是 MATLAB 中用于生成服从标准正态分布（均值为 0，方差为 1）的随机数的函数。

**2. wgn:**

- **wgn** 函数用于生成带有特定功率的高斯白噪声信号。它可以生成指定长度和功率的高斯白噪声信号，以用于模拟通信系统中的噪声。

**3. xcorr:**

- **xcorr** 函数用于计算两个序列之间的互相关（cross-correlation）。在信号处理中，互相关用于衡量两个信号之间的相似性或相关性。

**4. autocorr:**

- **autocorr** 函数用于计算信号的自相关（autocorrelation）。自相关是信号与其自身在不同时间延迟下的相关性。对于高斯白噪声，自相关通常在零延迟处达到峰值，并且在其他延迟值上接近零，

在 MATLAB 的 `autocorr` 函数中，滞后阶数是非负整数。这意味着它默认只会计算从滞后 0 开始的自相关函数值。因此，对于负数滞后，你需要手动计算自相关函数。

在 MATLAB 中，如果你想要计算自相关函数，你需要确保指定的滞后阶数不超过观测值的数量减去 1。你可以根据你的观测值数量来设置合适的滞后阶数，确保不超出范围。比如，如果你有 1000 个观测值，你可以设置最大的滞后阶数为 999。