

Github 账号: bleaner**实验摘要:**

- 熟悉 Matlab 软件平台和基本操作;
- 掌握利用 Matlab 来显示常用信号波形;
- 掌握利用 Matlab 来实现信号的时域变换。

实验题目

1. 利用MATLAB实现下列信号，并绘出图形

- (1) $f_1(t) = \varepsilon(t)$ ，取 $t = -1 \sim 10$
- (2) $f_2(t) = 4e^{-0.5t} \cos(\pi t)$ ，取 $t = 0 \sim 10$
- (3) $f_3(t) = g_2(t) + g_4(t)$ ，取 $t = -10 \sim 10$
- (4) $f_4(k) = \varepsilon(k+2) - \varepsilon(k-5)$
- (5) $f_5(k) = 7(0.6)^k \cos(0.9\pi k)$
- (6) $f_6(t) = Sa(t) = \sin(t)/t$

2. 利用MATLAB实现以上信号 $f_3(t)$ 的变化:

- (1) $f_3(2t)$
- (2) $f_3(4-2t)$
- (3) $f_3'(4-2t)$

3. 如图:

9. *** 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 positive scalar 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 $n == 200$ or greater and plotting the result, and you will see why the function is called "square_wave".

实验内容

一、实验原理

1.plot

绘制二维图形的最基本函数，它是针对向量或矩阵的列来绘制曲线的。也就是说，使用 plot 函数之前，必须首先定义好曲线上每一点的 x 及 y 坐标，或者是形如 $y=f(x)$ 这样确切的表达式。

常用格式为：

(1) plot(x) %当 x 为一向量时，以 x 元素的值为纵坐标，x 的序号为横坐标值绘制曲线。当 x 为一实矩阵时，则以其序号为横坐标，按列绘制每列元素值相对于其序号的曲线，当 x 为 $m \times n$ 矩阵时，就有 n 条曲线。

(2) plot(x,y) %以 x 元素为横坐标值，y 元素为纵坐标值绘制曲线。

(3) plot(x,y1,x,y2,...) %以公共的 x 元素为横坐标值，以 y1,y2,... 元素为纵坐标值绘制多条曲线。

2.stem

绘制离散序列数据

常用格式为：

stem(Y) %将数据序列 Y 绘制为从沿 x 轴的基线延伸的针状图。各个数据值由终止每个针状图的圆指示。

stem(X,Y)%在 X 指定的值的位置绘制数据序列 Y。X 和 Y 输入必须是大小相同的向量或矩阵。另外，X 可以是行或列向量，Y 必须是包含 length(X) 行的矩阵。

stem(____,'filled') % 填充圆形。可以将此选项与前面语法中的任何输入参数组合一起使用。

3.diff

差分 and 近似导数。

常用格式：

$Y = \text{diff}(X)$ %

%计算沿大小不等于 1 的第一个数组维度的 X 相邻元素之间的差分：

- 如果 X 是长度为 m 的向量，则 $Y = \text{diff}(X)$ 返回长度为 m-1 的向量。Y 的元素是 X 相邻元素之间的差分。

$$Y = [X(2)-X(1) \quad X(3)-X(2) \quad \dots \quad X(m)-X(m-1)]$$

- 如果 X 是不为空的非向量 $p \times m$ 矩阵，则 $Y = \text{diff}(X)$ 返回大小为 $(p-1) \times m$ 的矩阵，其元素是 X 的行之间的差分。

$$Y = [X(2,:)-X(1,:); \quad X(3,:)-X(2,:); \quad \dots \quad X(p,:)-X(p-1,:)]$$

- 如果 X 是 0×0 的空矩阵，则 $Y = \text{diff}(X)$ 返回 0×0 的空矩阵

$Y = \text{diff}(X,n)$ %通过递归应用 $\text{diff}(X)$ 运算符 n 次来计算第 n 个差分。在实际操作中，这表示 $\text{diff}(X,2)$ 与 $\text{diff}(\text{diff}(X))$ 相同。

$Y = \text{diff}(X,n,\text{dim})$ 是沿 dim 指定的维计算的第 n 个差分。 dim 输入是一个正整数标量。

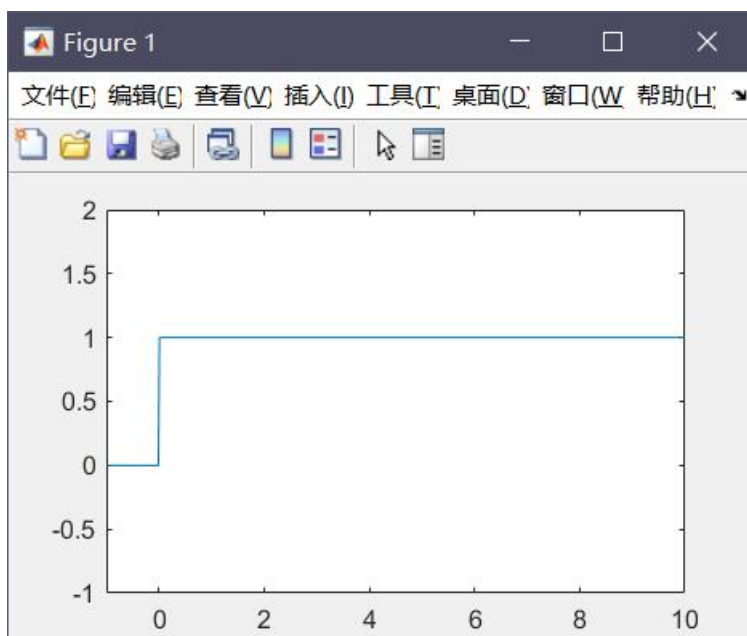
应用：求近似倒数

使用 diff 函数和语法 $Y = \text{diff}(f)/h$ 求偏导数近似值，其中 f 是函数值在某些域 X 上计算的向量， h 是一个相应的步长。

二、实验代码及结果

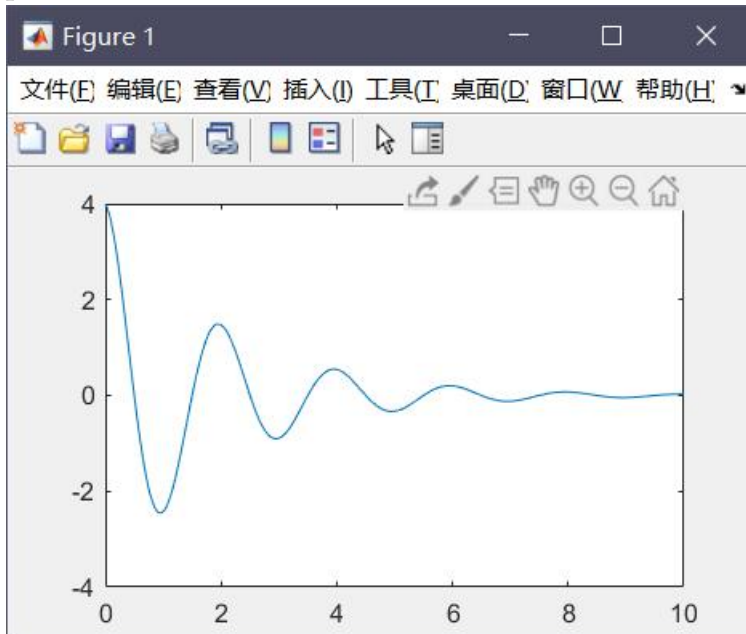
1.1

```
t = (-1:0.01:10);  
f1 = heaviside(t);  
plot(t,f1);  
axis([-1 10 -1 2])
```



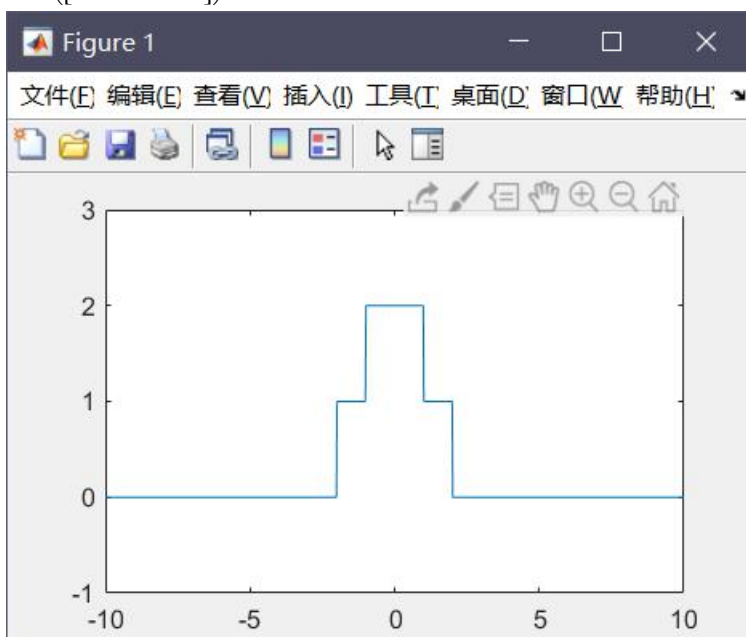
1.2

```
t = (0:0.01:10);  
f2 = 4 * exp(-0.5*t) .* cos(pi*t);  
plot(t,f2)
```



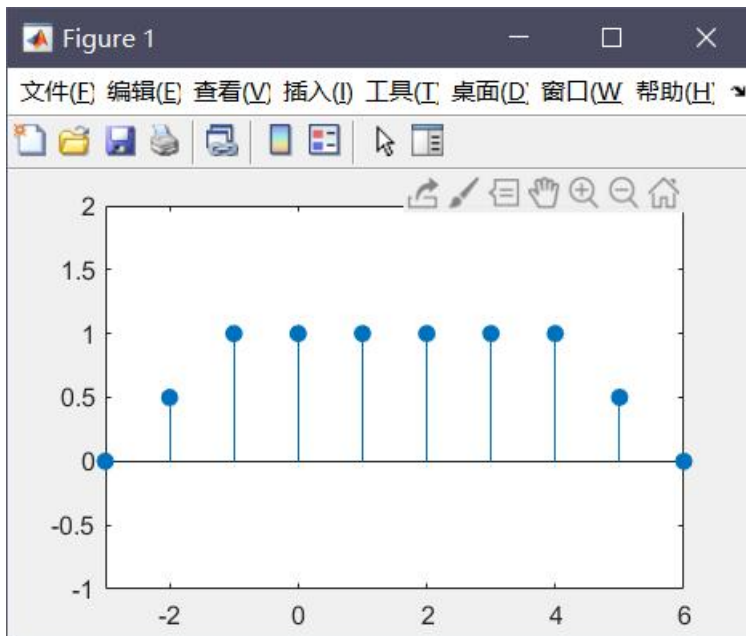
1.3

```
t = (-10:0.01:10);  
f3 = heaviside(t+1) - heaviside(t-1) + heaviside(t+2) - heaviside(t-2);  
plot(t,f3);  
axis([-10 10 -1 3])
```



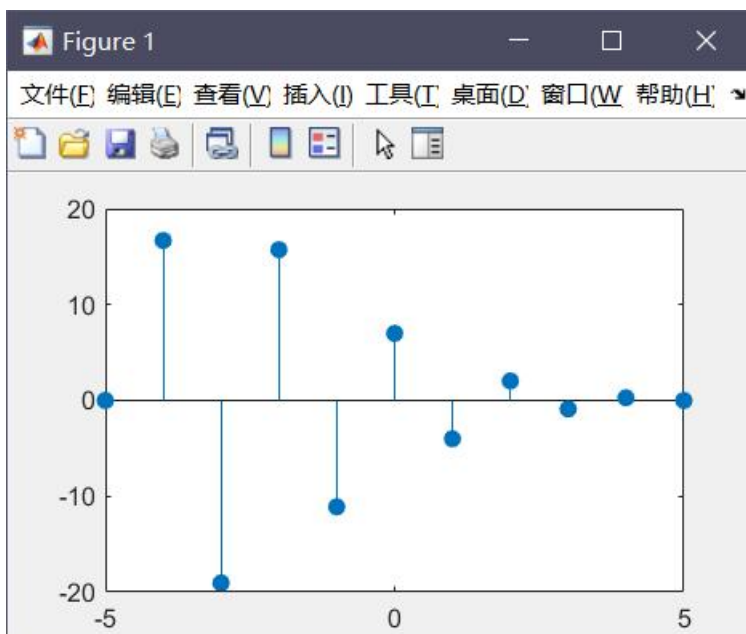
1.4

```
k = (-3:1:6);  
f4 = heaviside(k+2) - heaviside(k-5);  
stem(k,f4,'filled');  
axis([-3 6 -1 2])
```



1.5

```
k = (-5:1:5);  
f5 = 7 * (0.6.^k) .* cos(0.9 * pi * k);  
stem(k,f5,'filled')
```

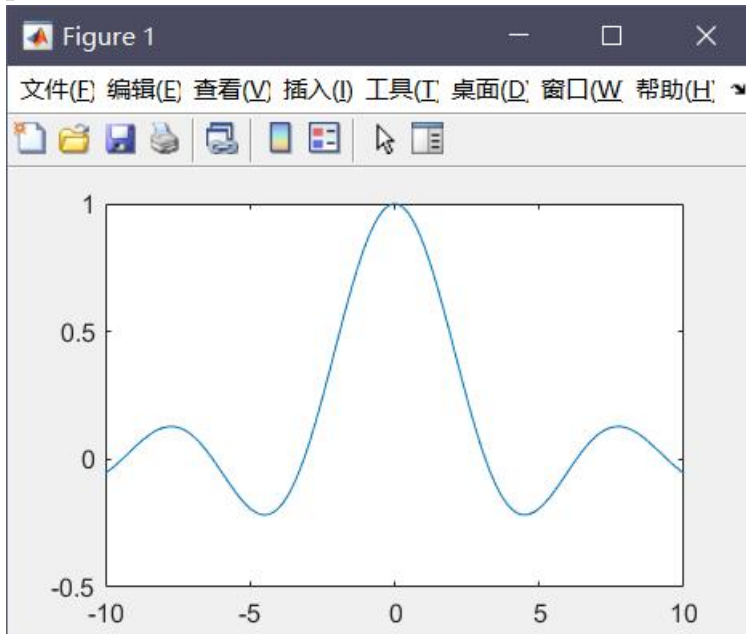


1.6

```
t = (-10:0.01:10);
```

```
f6 = sin(t) ./ t;
```

```
plot(t,f6)
```



2.

```
%f=f3
```

```
step = 0.01;
```

```
f = @(t) heaviside(t+2) + heaviside(t+1) - heaviside(t-1) - heaviside(t-2);
```

```
subplot(2,2,1); fplot(f,[-5, 5]); title("Figure f3(t)");
```

```
%f1=f(2t)
```

```
t0 = -5 : step : 5;
```

```
t = 2 * t0;
```

```
f1 = f(t);
```

```
subplot(2,2,2); plot(t0,f1); title("Figure f3(2t)");
```

```
%f2=f(4-2t)
```

```
t = 4 - 2 * t0;
```

```
f2 = f(t);
```

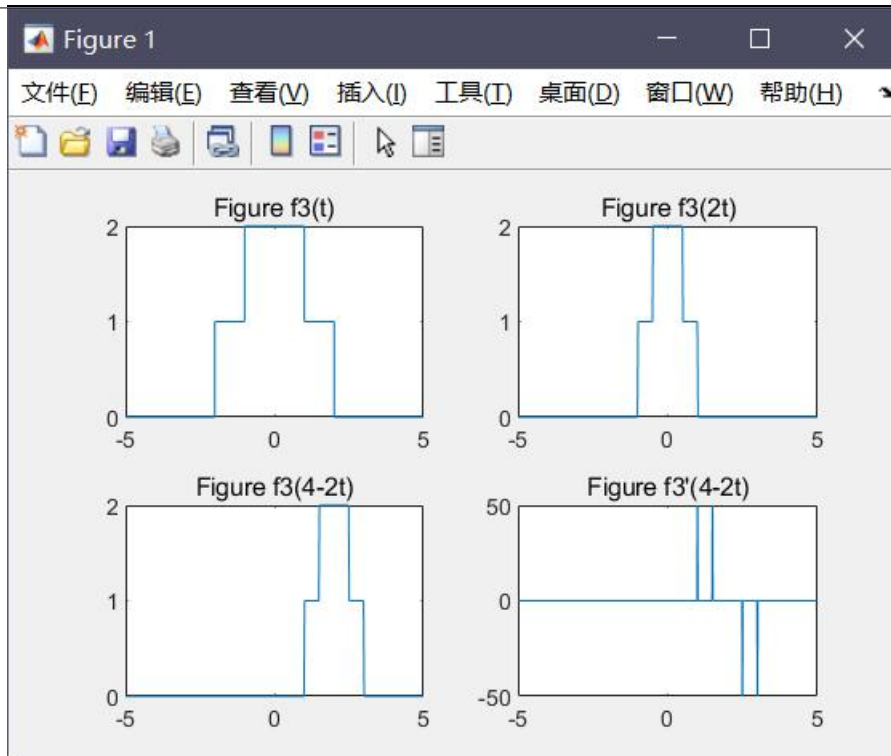
```
subplot(2,2,3); plot(t0,f2); title("Figure f3(4-2t)");
```

```
%f3=f'(4-2t)
```

```
t = t0;
```

```
f3 = diff(f2) / step;
```

```
subplot(2,2,4); plot(t(:,1:length(f3)),f3); title("Figure f3'(4-2t)");
```



3.

```
%square_wave.m
```

```
function sum = square_wave(n)
```

```
    t = linspace(0, 4*pi, 1001);
```

```
    sum = zeros(1,1001);
```

```
    for k = 1 : n
```

```
        sum = sum + sin((2*k-1)*t) / (2*k-1);
```

```
    end
```

```
end
```

```
%square_wave_test.m
```

```
subplot(3,1,1)
```

```
sum1 = square_wave(10);
```

```
plot(sum1);
```

```
title("Figure square_wave(10)");
```

```
subplot(3,1,2)
```

```
sum2 = square_wave(200);
```

```
plot(sum2)
```

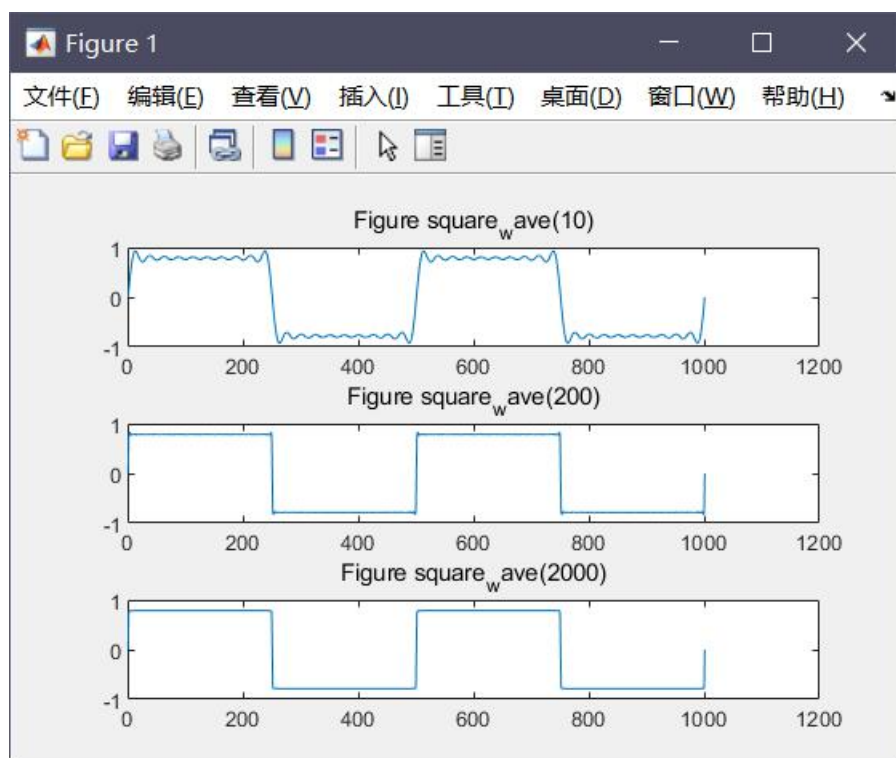
```
title("Figure square_wave(200)");
```

```
subplot(3,1,3)
```

```
sum3 = square_wave(2000);
```

```
plot(sum3);
```

```
title("Figure square_wave(2000)");
```



实验总结

- Matlab 函数的用法
- Matlab 中*与.*的区别

参考文献

- Matlab 官方文档
- <https://wenku.baidu.com/view/d366d2be1a37f111f1855b17.html>